The Automated Processing System

Naval Research Laboratory, Stennis Space Center, MS

Paul Martinolich

The Automated Processing System

Naval Research Laboratory, Stennis Space Center, MS
Paul Martinolich

Version 6.4.4.

Publication date 2016-08-30

Table of Contents

List of Figures

List of Tables

List of Examples

Part I. User's Guide

The chapters in Part I form a User's Guide which describes the Automated Processing System.

The Automated Processing System (aps) is designed to generate map-projected image data bases of satellite derived products from a large flow of raw satellite input data in an automated fashion. Individual scenes are sequentially processed from the raw digital counts (Level-1) using standard parameters to a radiometrically and geometrically corrected (Level-3) product within several minutes. It further processes the data into several different temporal (daily, 8-day, monthly, yearly, and latest pixel) composites or averages (Level-4). These products are stored in the Hierarchical Data Format (HDF) with aps specific attributes. Additionally, it automatically generates quick-look “browse” images. All processing is controlled through a data base (MySQL).

The aps was designed for the Naval Research Laboratory Remote Sensing Applications Branch at Stennis Space Center, MS (now the Ocean Sciences Branch) to handle the continuous stream of satellite data. Originally, the system was designed to produce sea surface temperature maps from data collected by the Advanced Very High Resolution Radiometer (AVHRR). Data from the AVHRR was received daily (up to six passes per day). The aps provides for near real-time processing with the option of reprocessing historical data. The aps has since been upgraded to process ocean color satellite data from: GOCI, MERIS, MODIS, SeaWiFS and VIIRS.

The aps uses a data base (SQLite, MySQL, or Postgres) with some user configuration XML file(s) to control processing. All user-provided input data must be registered with the data base and all data created by aps will be registered. This data base is queried by aps as needed.

Through the use of several XML configuration files the user can control which sensors are processed. One may only process data that covers a specified region of interest during a specific time frame. One may control which Level-2 products are generated and which processing parameters are used (cloud albedo, atmospheric correction method, etc.). One may control which images and how they are generated (full resolution, number of grids and their color, etc.).

The 6.4.4 version represents the processing algorithms employed at the Naval Research Laboratory as of 2016-08-30 for the GOCI, MERIS, MODIS, SeaWiFS and VIIRS sensors. The system has been developed on Scientific Linux 6.7 (x86, x86_64).

Table of Contents

Chapter 1. Introduction to the Automated Processing System Concept

The Automated Processing System (APS) is a system designed to allow scientists to generate co-registered image databases of geophysical parameters derived from remotely sensed data.

The APS does not contain any GUIs or visualization programs. Therefore, to address these characterizations, all user input *must* be provided to the program upon *start*.

Understanding the APS

The APS is a system designed to process ocean color satellite data. Specifically, the system contains procedures for:

- atmospheric correction
- geometric correction
- image processing
- compositing
- databasing
- vicarious calibration
- acquisition of input

Chapter 2. Getting Started

System Requirements

The system requirements are difficult to gauge. The amount of memory needed is dependent upon the amount and type of satellite data you wish to process. The larger the area, the larger the requirements. For example, the entire Atlantic Ocean will require more processing power than the Mississippi Bight. Plus, the type of data being processed will determine how robust the system should be.

Likewise, the amount of disk space is dependent upon the amount and type of satellite data one wishes to process. The requirements for one-year of SeaWiFS data over the Gulf Of Mexico are much different than those for the entire MODIS/Aqua mission for the globe.

Finally, the selection of the database is important also. For a small single processor system for a small region and time frame, the default SQLite database should be sufficient. However, as the number of processing threads increase, the non-concurrent nature of SQLite begins to cause issues. In normal processing, it is best to use a PostgreSQL database.

Installation

Note

If APS has already been installed, please skip to the next section.

Once the system requirements have been met, you should be ready to install APS. APS is distributed as a series of UNIX tar files. This should be extracted from the user's `$HOME` directory.

The default parameters used by the APS are internally generated and should suffice for most users. However, the user can provide an XML configuration file that changes these defaults. See the documenation on the **aps.rb** for an in-depth examination of the APS configuration parameters.

Prior to starting APS it is highly recommended that one examines these parameters by issuing the APS command **config verbose**. This will print to the screen, the current run parameters. It provides a list of directory locations for where all input Level-1 and generated output files and images will be located as well as processing parameters (number of threads, location of SQL data base, etc).

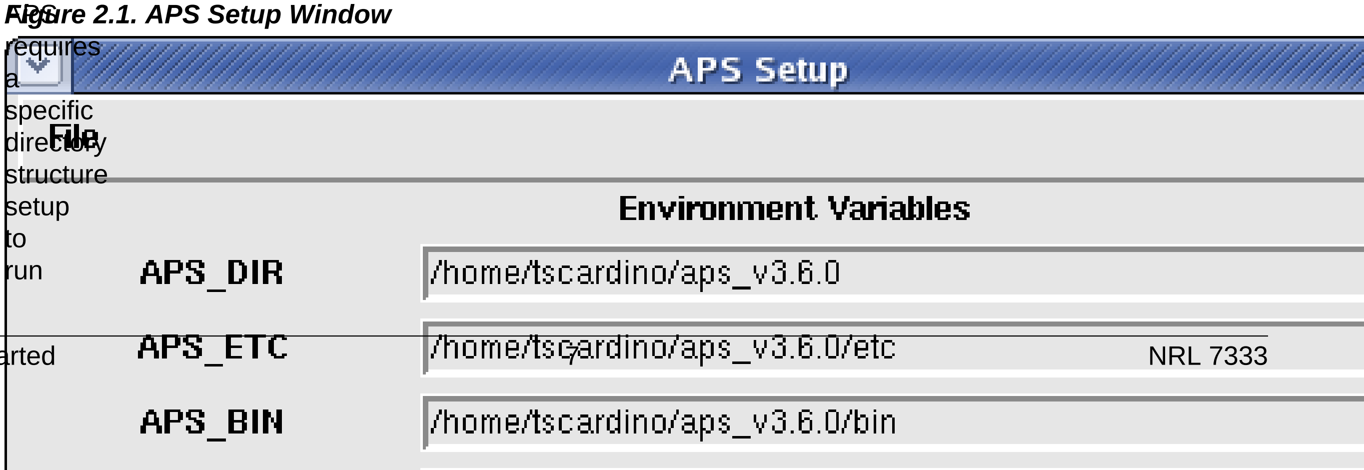
GUI Installation

To begin the program, run the `aps.rb` executable. It is located in the `$HOME/aps_6.4.4/bin/` directory. When the APS interface starts you will first see the Setup window. APS uses a default setting for everything included in this window. However, your installation may have some differences that need to be addressed before running APS. Here we will review the APS Setup window and look at what each field means.

APS Setup Window

Figure 2.1. APS Setup Window

APS requires a specific directory structure setup to run



correctly. The top-level directory is `aps_v6.4.4`. Under this directory are the `bin`, `lib`, `data`, `in`, `work`, `out`, `err`, and `var` subdirectories. Since every directory is used by the system, the default values are generated. However, the Setup window allows the user to change that according to his/her unique system setup. This becomes very important when the user is not the owner of the installation executable. Usually, the default variables and parameters are acceptable. Simply review the window, make the appropriate changes and select `Save` from the `File` menu.

Environment Variables

The environment variables keep track of the directory locations. These variables are defined based on the user's perspective, which one installation might have multiple users.

APS_DIR: Top-level directory of APS.
APS_BIN: Location of APS binaries.
APS_ETC: Site configuration files.
APS_DATA: The data for APS.
APS_LIB: Location of APS libraries and scripts.
APS_WORK: While APS is processing it will use this directory to work in.
APS_OUT: The completed products will be in this directory.
APS_ERROR: If an error is thrown, this directory will contain the log messages.
APS_VAR: Variable data.
APS_VAR_DATA: Dynamic data that requires many updates, like calibration files.
APS_LOCK: A common directory where most APS lock files will be found.
APS_LOG: A common directory that holds the log files.

Database Variables

APS_DATA_BASE: Allows browsing of the rs directory for data.
APS_L1_DATA_BASE thru APS_L5_DATA_BASE: Output data locations with levels of 1-5 data, respectively.
APS_IMAG_BASE: Allows browsing of the browse directory for data.
APS_L1_IMAG_BASE thru APS_L5)IMAG_BASE : Output data locations with levels of 1-5 data, respectively.

Processing Parameters

Verbosity: The default is set to '1', which means 'on'. This tells the system to report more information out as it runs and be verbose.
Number of Threads: The default is set to '1'. Typically, this number is set to the number of CPUs on your system due to the multi-threading capability of the CPU.

File Parameters

Log File: The location of the log file.
PID File: The location of the PID (Process Identification) file.
Process File: The location ofthe process file.

Chapter 3. Using the APS

To get started using the APS, we will create the necessary configuration files for the creation of chlorophyll-a image maps for waters within the Gulf of Mexico. It is assumed that the APS has been installed at your site. If the system has not been installed, see the section called “Installation” Furthermore, it is assumed that APS was installed in `$HOME/aps_v6.4.4` and that the variable `APS_DIR` is set to `$HOME/aps_v6.4.4`.

Defining Your Interest

Before creating the necessary configuration files, we must decide what it is we want to create. In this example, the objective is to create an image data base of chlorophyll a concentration using two different algorithms for waters within the Gulf of Mexico. We wish to compare the default chlorophyll-a algorithm (OCI) with the OC3 algorithm in both open ocean and coastal waters. We plan to use the data from the VIIRSN sensor.

Waters Within the Gulf of Mexico

First, we will define our region of interest. For this example, it will be defined as the waters of the Gulf of Mexico. Using a USGS or National Geographic map of the Gulf of Mexico, we decide to define this region as the area within the following geographical boundaries:

- The upper left corner shall be 31 N and 98 Wn
- The lower right corner shall be 17 N and 80 W

Because we plan to do spatial and temporal variability studies of the loop current in the Gulf of Mexico (and maybe make a nice movie loop of several days worth of images), we need each scene to be registered to a standard map projection. We use the term *image map* to indicate a specific map projected image. An *image map* is the combination of a map projection and an output image. Each pixel's relationship to its neighbors is defined by the projection used. In this example, we want our image to be a full-resolution image of chlorophyll a represented as a Mercator projection. The Mercator projection as one of thirty available map projections provided by the General Cartographic Transformation Package. The GCTP package was obtained from the United States Geological Survey.

One group of outputs generated by the Level-2 APS programs is collectively termed a control points structure. This structure allows a program to determine the geographical coordinates of any pixel in the original scene by tying down specific points in the satellite scene (sample,line) to geographical coordinates (longitude,latitude). These points are commonly called Ground Control Points (GCP). In the APS, the structure forms a regular grid across the image. That is, the for each column and row in the input image, a known latitude and longitude pair is given. Thus, the column or pixel locations as well as the row or line locations are given as 1-dimensional arrays. The pixel and line locations are stored in separate arrays. The latitudes and longitudes at the intersections of the row and columns are stored in 2-dimensional arrays. The latitudes and longitudes are also stored in separate arrays. This information is used by the geometric registration program.

The APS includes a geometric registration program known as **imgMap**. This program moves the pixels or data from the input file to newer locations in the output file. This process is known as geometric registration or “warping”. The translation from input pixel to output pixel is defined by the *image map* and the control points structure. The results of the pixel movement will depend up the accuracy of the input control points structure. Additionally, the parameters unique to the selected map projection will effect the accuracy in scale of the resulting image. For example, the Mercator map projection will exaggerate land mass as we reach the poles. That is, Greenland will appear much larger in size than it actually is.

Controlling the Projection Characteristics of the Output

The

Figure 3.1. APS Map Example

image map parameters determine the projection of the output image as well as of the imaginary map of the world of which it is a part. Although the image is projected onto that map, the input parameters allow the user to control where the window will be positioned over the map as well as the scale of the map.

The first two required parameters will associate a longitude and latitude (LL_1) from the world to a particular location on the output image (XY_1). The XY_1 parameter refers to the image coordinates (sample and line location) of a point in the image whereas LL_1 refers to the geo-coordinates (longitude and latitude) of a point on the world map. The points will be associated so that the image point (XY_1) overlays the map point (LL_1). It may be convenient to choose LL_1 to be the center of the image by assigning XY_1 to the first coordinate :number of samples/2, and the second coordinate: number of lines/2. Alternately, the user may wish that a certain landmark appear at a certain location in the image. In such a case, the landmark's geo-coordinates would be entered for LL_1 and the desired image location specified by XY_1.

The next two parameters will determine the scale of the *image map*. The first parameter is a second point on the world map (LL_2) and the second parameter is a distance (in pixels) away from the first (DELTA). The use of LL_2 and DELTA in conjunction with LL_1 and XY_1 controls the scale of the map and, hence, also controls how much of the mapped image appears within the image. LL_2 represents the geo-coordinates of another point on the world map and DELTA represents the separation in pixels between that point and the XY_1 image location. A positive DELTA represents a horizontal separation, whereas a negative value represents a vertical separation.

Note that this second point need not be within the image and that the absolute value of DELTA may be larger than the image width or height. For given parameters, a larger absolute DELTA will decrease the geographical area covered by the window (enlarge the map or *increase* the scale); a smaller absolute value will increase this area (contract the map or *decrease* the scale). The direction of the second point relative to the first - that is, where they both fall on the world map - is determined solely by the projection.

Although DELTA represents the separation in either the horizontal and vertical direction (not the absolute separation), the points for LL_1 and LL_2 must be chosen such that they have both a horizontal *and* vertical separation on the imaginary map. Therefore, some *a priori* knowledge of where these points will fall on that map is required when choosing these parameter values.

A convenient way to determine the value for LL_2 and DELTA is to use the geo-coordinates of another landmark for LL_2 and enter the desired separation between LL_1 and LL_2 for DELTA. Another convenient way to determine these values is to determine the scale for the map at LL_1. (In certain cases, depending on the projection, the scale will vary greatly even within the image area.) That is, the user decides how many image pixels (DELTA) should separate a longitudinal or latitudinal degree and assign LL_2 accordingly. For example, if the scale at LL_1 is to be one latitudinal degree per 100 pixels and LL_1 is 10 degrees latitude and 38 degrees longitude, LL_2 would be 9 and 38 degrees and DELTA would be -100 (assuming that north is on the top of this projection).

The example below may help understand this idea. The window is represented by the red box with the points labeled to show XY_1, LL_1, and LL_2. The arrows show the DELTA.

Creating the Map

In this example, we align the North West corner (LL_1) with the upper left corner of the image (XY_1). Thus, LL_1 will have a value of 98 W and 31 N and XY_1 will have a value of sample 1, line 1. (For images produced by **imgMap** the origin of the image is located in the upper left. In other words, we count our lines down from the top and our samples (or columns) from the left. So, line 1 is at the top of the image; line n is at the bottom. Pixel 1 is on the left edge of the image; pixel n is on the right edge.)

To define the second point (which does not have to exist in the output image), we need to set the DELTA component. This component is the horizontal or vertical separation in pixels on the projected image between the LL_1 (the NW corner) and LL_2. This parameter, along with the latitude and longitude of the second point, define the resolution of the output image. To maintain a 0.01 degrees/pixel resolution in the longitudinal direction, we set DELTA to 1799. If pixel 1 corresponds to 98 W and pixel 1800 corresponds to 80 W, then there are 1800 - 1, or 1799, pixels between them.

For the time being we will assume that the scale in the latitudinal direction is the same as in the longitudinal direction, though for the Mercator projection this is certainly *not* true. Since we want to place LL_2 near 80 W and 17 N, we will start by making a temporary map with 1400 lines because the Northern and Southern latitude differ by 14 degrees and we have a scale of 0.01 degrees/pixel.

Once the map has been defined we will use the **toll** command to get a better approximation. Remember that LL_1, XY_1, LL_2, and DELTA define the image to the imaginary map with LL_1 and XY_1 being tied together. The size of the window (our output image) over the imaginary map can be different in size

The program **maps** is used to create the *image map*. We give the mapped image a name and store the parameters in an HDF file for later use. We call this file *image map*. Note that in the example below, we set the central meridian and latitude of true scale to the middle of our projection and that we use the WGS 84 datum (12). See maps(1) for a complete list of possible map projections and datums.

So we begin by running the **maps** program and attempt to create our first *image map*:

```
$ cd
$ pwd
/home/aps/
$ maps
maps>create temp
Full Name? [ temp ]
Code (3-letter)? [ temp ]
Map Projection (? for list)? [ Mercator ]
Size of Map (w h)? 1800 1400
Pt1 of Map (lon lat)? -98.0 31.0
Pt1 of Map (x y)? [ 1 1 ]
Pt2 of Map (lon lat)? -80.0 17.0
Delta (pixels)? [ 1799 ]
Aspect ratio? [ 1.0 ]
Datum (? for list)? [ WGS 84 ]
SMajor? [ 6378137.000000 ]
SMinor/Eccentricity? [ 6356752.314245 ]
Longitude of Central Meridian (lon)? -89.0
Latitude of True Scale (lat)? 24.0
False Easting? [ 0.0 ]
False Northing? [ 0.0 ]
maps>setmap temp
maps>toll 1 1
-98.000000 31.000000
maps>toll 1800 1400
-80.000000 18.241954
maps>toxy -80.0 17.0
1800.000000 1529.447826 - off map
```

From this last information, we will increase the size of the image (that is, the *window* over the imaginary map) to 1530. So to continue:

```
maps>create Gulf
```

```
Full Name? [ Gulf ] Gulf of Mexico
Code (3-letter)? [ Gulf ] GOM
Map Projection? [ Mercator ]
Size of Map (w h)? 1800 1530
Pt1 of Map (lon lat)? -98.0 31.0
Pt1 of Map (x y)? [ 1 1 ]
Pt2 of Map (lon lat)? -80.0 17.0
Delta (pixels)? [ 1799 ]
Aspect ratio? [ 1.0 ]
Datum (? for list)? [ WGS 84 ]
SMajor? [ 6378137.000000 ]
SMinor/Eccentricity? [ 6356752.314245 ]
Longitude of Central Meridian (lon)? -89.0
Latitude of True Scale (lat)? 24.0
False Easting? [ 0.0 ]
False Northing? [ 0.0 ]
maps>setmap
Name? Gulf
maps>toll 1800 1530
-80.000000 16.994684
maps>delete temp
maps>list
Gulf
maps>save gulf_map.hdf
maps>quit
$ ls gulf_map.hdf
gulf_map.hdf
$ hdf gulf_map.hdf list
File: gulf_map.hdf

File Attributes: (4)

createTime = "Fri Aug 18 20:04:43 2006"
createSoftware = "APS v6.4.4"
createPlatform = "i686-pc-linux-gnu"
createAgency = "Naval Research Laboratory, Stennis Space Center"

Data Sets: (1)

projection_t Gulf
    Name         Gulf
    Full Name    Gulf of Mexico
    Code         GOM
    Projection    5 (Mercator)
    Zone         62
    Datum        12 (WGS 84)
    Parameters:
     0:  6378137.000000 (Semi-Major Axis)
     1:  6356752.314245 (Semi-Minor Axis)
     2:  0.000000
     3:  0.000000
     4:  -89000000.000000 (Longitude of Central Meridian)
     5:  24000000.000000 (Latitude of True Scale)
     6:  0.000000 (False Eastings)
```

```
     7:  0.000000 (False Northings)
     8:  0.000000
     9:  0.000000
    10:  0.000000
    11:  0.000000
    12:  0.000000
    13:  0.000000
    14:  0.000000
    Width        1800
    Height       1530
    Longitude_1  -98.0000
    Latitude_1    31.0000
    Pixel_1      1
    Line_1       1
    Longitude_2  -80.0000
    Latitude_2    17.0000
    Delta        1799
    Aspect       1
```

To process this region copy the newly created map file `gulf_map.hdf` to the APS `etc` directory. By default, the APS has a default map file to hold all known maps named `default.maps` It resides in the `etc` directory. To use the default location, you can copy the `Gulf` map from the `gulf_map.hdf` file to the `default.maps` file. If not, then your configuration files will have to set the `file` variable so that APS will know which file to look at to find the `Gulf` projection information.

Note

See reference page for **maps** for additional examples.

To copy the `gulf_map.hdf` file to the `etc` directory use the command (remember we will have to add `MapFile` to our script):

```
cp gulf_map.hdf ~/aps_v6.4.4/etc
```

To copy the `Gulf` area from the `gulf_map.hdf` file to the standard APS maps file (`default.maps`), we will use the APS provided program **hdf**.

Note

The name we select should not already be in the `default.maps`.

```
hdf gulf_map.hdf copy ~/aps_v6.4.4/etc/default.maps Gulf
```

Level-1 to Level-2

Now that the projection information is done, we will start to create all the necessary configuration files to produce the products and images for this example.

To begin we start with the Level-1 to Level-2 processor. The **n2gen** program is responsible for this. The `$APS_ETC/l2_gen.xml` is the configuration file that we must create and/or modify.

A single run of **n2gen** is controlled by the XML nodes called <recipe>. Within this node the user may define a set of products or suites of products, various processing parameters, and other controls uing XML attributes. The basic <recipe> structure is defined below.

```
<recipe>
```

```
    <environment>
        <variable>key=value</variable>
    </environment>
    <timeframes>
        <timeframe>start,stop</timeframe>
    </timeframes>
    <params>
        <param>key=value</param>
    </params>
   [<suites>]
         [<suite>]
             <products>
                 [<day|night>]
                     <product>value</product>
                 [</day|/night>]
             </products>
         [</suite>]
   [</suites>]
 </recipe>
```

For our working example, we'll produce only the two chlorophyll-a products for VIIRSN. Since this sensor acquires data both during the day and nighttime, the node <day|night> is needed. Had we desired the same images from SeaWiFS, the node <day> would not be required. Therefore, our XML <recipe> node will look like:

```
<recipe>
    <products>
        <day>
            <product>chlor_a</product>
            <product>chl_oc3</product>
        </day>
    </products>
</recipe>
```

Now since the APS can run multiple satellites and multiple runs of **n2gen**, the `$APS_ETC/l2_gen.xml` file has an overall structure that will contain the various <recipe> nodes. The overall structure looks like:

```
<l2_gen>
    <viirsn>
        <recipes>
        </recipes>
    </viirsn>
 </l2_gen>
```

Note that the sensor node <viirsn> is specific to the name of the sensor. Therefore, the only valid node values here are: <goci>, <meris>, <modisa>, <seawifs>, and <viirsn>. Our final `$APS_ETC/l2_gen.xml` will be:

```
<l2_gen>
```

```
        <viirsn>
            <recipes>
                <recipe>
                     <products>
                         <day>
                             <product>chlor_a</product>
                             <product>chl_oc4</product>
                         </day>
                     </products>
                </recipe>
            </recipes>
        </viirsn>
    </l2_gen>
```

There are several options that control when a file should be run using the n2gen <recipe>. These are found as attributes to the <recipe> node.

The attributes are:

name="name"	used for informational purposes
active="yes"	used to turn on/off this recipe
daynight="day"	used to determine if recipe is specific to day-time passes or night time (used only for VIIRS and MODIS, ignored by other sensors)
resolution=xxx	used to determine if recipe is specific to a sensor resolution (use only for VIIRS, MODIS, and MERIS)
recipe="string"	passed to n2gen as "recipe=string" for naming conventions useful when processing same Level-1 with different atmospheric options
area="string"	passed to n2gen as "area=sstring" for naming conventions useful when processing a subsection of the Level-1 file (local)
file="path"	used to define the file which contains the map defined in map attribute
map="region"	used to determine if recipe will be executed based on coverage useful when processing a subsection of the Level-1 file (local)
archive=regexp	used to define the desired archive location if not set (best option), the bin/RSDB -a option will determine default location
ancil=""	used to set a list of ancil data to use with this recipe may be: seaice, met, ozone, no2, vcm

Level-2 to Global “Binned” Product

The daily Level-2 images can be combined into an 11km world map projection using the **imgMap** program. The `$APS_ETC/global.xml` is the configuration file that we must create and/or modify. For example:

```
<global>
     <active>yes</active>
     <map>Hammer90W</map>
```

```
     <sensors>
         <sensor name='viirsn'>
             <products>
                 <product>chlor_a</product>
                 <product>chl_oc3</product>
             </products>
             <images>
             </images>
         </sensor>
     <sensors>
 </global>
```

With the current version of APS, the <map> node may only be set to `Hammer90W`. The list of senors may only include `modisa` or `viirsn`. The products in the <products> node should only include products that are generated for all Level-2 files.

The optional section <images> can be set to define which images are created and the options used. See the section below for more details about image creation.

Level-2 to Regional Level-3 and Level-4 Products

The Level-2 files may also be projected into several regional map projections like the one we defined above. To define the set of regions, APS uses the `$APS_ETC/regions.xml` configuration file. This XML file contains information about which regions are active, for which sensors and provides some bounding coordinates to aide APS to quickly determine the Level-2 file coverages.

```
<regions>
     <region name='Gageocho_Station'>
         <active>yes</active>
         <type>AERONET-OC</type>
         <sensors>
             <sensor>goci</sensor>
             <sensor>modisa</sensor>
             <sensor>viirsn</sensor>
         </sensors>
         <north>34.8</north>
         <south>33.2</south>
         <west>123.7</west>
         <east>125.5</east>
     </region>
     <region name='Gulf'>
         <active>yes</active>
         <level4>yes</level4>
         <sensors>
             <sensor>modisa</sensor>
             <sensor>viirsn</sensor>
         </sensors>
         <north>31.0</north>
         <south>17.0</south>
         <west>-98.0</west>
         <east>-80.0</east>
     </region>
```

```
</regions>
```

In this example, we have defined two regions. For VIIRS and MODIS Level-2 data, both regions will be processed. The GOCI is processed only for the first region. If we wish to stop processing a region, the <active> node can be set to “no”.

The bounding coordinates are passed down to the **apsAreas** program by APS. This allows that program to read the bounding coordinates of the input file and quickly determine if the file does not cover this region. This greatly speeds up the processing. If the input file does not contain bounding coordinate attributes, the file is processed normally.

For speed considerations, any region that will have composites (Level-4) generated should have <level4>yes</level4> set. APS will use this information to only examine those regions when processing the composites.

The <type> node may only have one value (AERONET-OC) if provided. This node is used to determine which regions will be processed by the vical processor **vc-aeronet.rb** provide that vicarious calibration processing is on.

Level-2 to Regional Level-3 Products

Next we look at the Level-2 to Level-3 processor. The **imgMap** program is responsible for this. The `$APS_ETC/l3_gen.xml` is the configuration file that we must create and/or modify.

A single run of **imgMap** is controlled by the XML nodes called <map>. Above this node the user may define a set of products and various processing parameters, and other controls uing XML attributes. The basic structure is defined below.

There are several attributes in the <map> node, of which, the **name** attribute is required. The others are used to as modifiers or contionals to processing. For example, the **zoom** attribute defines the -z option which is passed to **imgMap**. The **code** attribute can be used to change the default code in the Level-3 file name.

```
<l3_gen>
     <sensor>
         <options>imgMap options</options>
         <maps>
             <map name="mapName"
                 zoom="zoom factor"
                 code="code"
                 resolution="resolution"
                 match="regexp"
                 archive="/archive/path/"
                 file="path/to/map/file">
                 <timeframes>
                     <timeframe>start,stop</timeframe>
                 </timeframes>
             </map>
         </maps>
     </sensor>
 </l3_gen>
```

Note that the sensor node <sensor> is specific to the name of the sensor. Therefore, the only valid node values here are: <goci>, <meris>, <modisa>, <seawifs>, and <viirsn>.

The **resolution** and **match** attributes are used to filter the Level-2 files which are possible candidates for warping to Level-3. The **resolution** attribute can be set to the desired sensor resolutions which will be warped. For MODIS data, this can be used to set maps which zoom factors to use.

```
<l3_gen>
     <modisa>
         <maps>
             <map name="Gulf" resolution="1000" zoom="1.0"></map>
             <map name="Gulf" resolution="250" zoom="4.0"></map>
         </maps>
     </modisa>
</l3_gen>
```

The **match** attribute can be used to restrict which Level-2 files are warped based on a regular expression. For example, if we only wish to warp the VIIRS night-time SST products to a EastCoast projection:

```
<l3_gen>
     <viirsn>
         <options>-X draw=1,masks=TRIMPIXEL</options>
         <maps>
             <map name="EastCoast" match="N.L2_SST"></map>
         </maps>
     </viirsn>
</l3_gen>
```

The **file** attribute should be used if the map given in name is not in the default map file `$APS_ETC/default.maps`.

```
<l3_gen>
     <viirsn>
         <options>-X draw=1,masks=TRIMPIXEL</options>
         <maps>
             <map name="MyMap" file="/home/user/my.maps"></map>
         </maps>
     </viirsn>
</l3_gen>
```

For our example, the `$APS_ETC/l3_gen.xml` would look like:

```
<l3_gen>
     <viirsn>
         <options>-X draw=1,masks=TRIMPIXEL</options>
         <maps>
             <map name="Gulf"></map>
         </maps>
     </viirsn>
</l3_gen>
```

Note that since the VIIRS sensor has samples in the file that were aggregated by the on-board sensors (marked by the TRIMPIXEL bit), we use the **-X draw=1,masks=TRIMPIXEL** option passed to **imgMap** to handle these samples.

Regional Level-3 to Regional Composited Level-4

Next we look at the Level-3 to Level-4 processor. The **imgMean** program is responsible for this. The `$APS_ETC/l4_gen.xml` is the configuration file that we must create and/or modify.

A single run of **imgMean** is controlled by the XML nodes called <composite>. Within this node the user may define a set of products or suites of products, various processing parameters, and other controls uing XML attributes. The basic <composite> structure is defined below.

```
<composite>
     <options>imgMean options</options>
     [<suites>]
           [<suite>]
                 <products>
                     [<day|night>]
                         <product>value</product>
                     [</day|/night>]
                 </products>
           [</suite>]
     [</suites>]
</composite>
```

Now since the APS can run multiple satellites and multiple runs of **imgMean**, the `$APS_ETC/l4_gen.xml` file has an overall structure that will contain the various <composite> nodes. The overall structure looks like:

```
<l4_gen>
     <viirsn>
         <composites>
         </composites>
     </viirsn>
</l4_gen>
```

Note that the sensor node <viirsn> is specific to the name of the sensor. Therefore, the only valid node values here are: <goci>, <meris>, <modisa>, <seawifs>, and <viirsn>. Our final `$APS_ETC/l4_gen.xml` will be:

```
<l4_gen>
     <viirsn>
         <composites>
             <composite type="daily">
                 <options>-T 1 -m HISATZEN,NAVWARN,NAVFAIL</options>
                 <products>
                     <day>
                         <product>chlor_a</product>
                         <product>chl_oc4</product>
                     </day>
                 </products>
```

```
            </composite>
        </composites>
    </viirsn>
</l4_gen>
```

Images

Next we look at how images are created from the above products. The **imgBrowse** program is responsible for this. The basic XML structure for images is the following nodes:

```
<images>
    <options>overal options for all images</options>
    <image>
        <product>product</product>
        <options>options specific to this image<options>
        <alias>output filename product<alias>
        <format>image format <format>
        <match>filename regexp<match>
    </image>
</images>
```

The <options> nodes provide for options passed to **imgBrowse** for all images and specific images. The format can be set (default is png). The <alias> can be used to replace the product name with the alias name in the output file nomenclature. The <match> node can be used to specify that this image will only be created for files which match the given regular expression.

For Level-2 files, this node is embedded inside a <sensor> node with is embedded in the top-level <l2_browse> node. The configuration file for Level-2 images is `$APS_ETC/l2_browse.xml`. It has the following structure.

```
<l2_browse>
    <viirsn>
        <images>
            <image>
            </image>
        </images>
    </viirsn>
</l2_browse>
```

For the Level-3 and Level-4 files, simply repace the “l2” with “l3” and “l4”.

Note that the sensor node <viirsn> is specific to the name of the sensor. Therefore only the only valid node values here are: <goci>, <meris>, <modisa>, <seawifs>, and <viirsn>.

Our final `$APS_ETC/l3_browse.xml` will be:

```
<l3_browse>
    <viirsn>
        <images>
```

```
            <options>-p $APS_ETC/jpss.symbols</options>
            <image>
                <product>chlor_a</product>
            </image>
            <image>
                <product>ndvi</product>
                <options>-M draw=0 -l draw=0<options>
            </image>
            <image>
                <product>rhos_670:rhos_555:rhos_443</product>
                <options>-T long_name=True_Color -M draw=0 -l
draw=0<options>
                <alias>true_color<alias>
            </image>
        </images>
    </viirsn>
</l3_browse>
```

The results

After this completes, we can examine the browse images by changing to the directory created by APS for this. If we assume that the `APS_IMAG_BASE` variable was set to `/home/aps/browse` in the `aps.conf` file, then we can examine the Level-3 JPEG files with (or use any other graphics display program):

```
$ cd ~/browse/lvl3/seawifs/5.4/Gulf/2001/jan
$ display S2001005175131.L3_HNAV_GOM_chl_oc2.jpg
```

Where did that directory come from? APS creates it automatically. By default, APS will put all the HDF product files in the `rs` directory and the JPEG images in the `browse` directory. If the defaults were selected during the installation, this would be subdirectories of your home account. That is, `/home/aps/rs` and `/home/aps/browse`.

The remaining sub-levels are created automatically by APS. They follow the pattern: `level/sensor/area/year/month/`. This pattern is true for all Level-3 data (HDF or browse images). For Level-4, the basic pattern is the same, except we add a name to for the temporal nature of the composite and delete some subdirectories that are not needed. So, for Level-4, the directories are created for example: `lvl4/seawifs/Gulf/daily/2001/jan`, `lvl4/seawifs/Gulf/weekly/2001`, or `lvl4/seawifs/Gulf/yearly`.

You will notice that the SeaWiFS data is stored in a `v08` as part of the filename. This version is a *algorithm processing* version and may or may not reflect that software version number. This was done because the APS can handle many different sensors, whose processing algorithms may or may not change with the software versions. For APS v6.4.4, the SeaWiFS data is at version v10, the MODIS data is at version v11, the MERIS data is at version v06, the GOCI data is at version v01, and the VIIRS data is at version v02.

Thus far, we have been able to use the APS software to run the standard SeaWiFS processing for our area of interest. If we have only a few files, it would make sense to run the above commands for each SeaWiFS file. However, the strength of APS is that we don't just have a few files - we have hundreds or thousands of files. Or, we have a receive station that is continually capturing more data. Then running them by hand does not make much sense. The next section will show how to start processing the data automatically.

Another option for users who do not require automation (because you order and download a bunch of files from the DAAC and then just process them), is the **repro** option for the **aps** script. This option allows the user put a list of files into a UNIX text file and process only those files and areas that they wish.

Chapter 4. APS Site Configuration

This section will cover some of the other standard procedures deployed in a fully operational system, like changing your desired list of standard products, and changing the location of the database, etc. All site-specific files should go into the `APS_ETC` directory.

Changing the Logo and text string on the browse image

The browse images contain a logo in the bottom right corner and also three text strings which can be user defined. The strings on the left can not be modified. The logo displayed is read from the JPEG formatted file `$APS_ETC/logo.jpg`. The user should name their logo as such. The logo file should generally be small about 48 by 48 pixels. The APS software will not reduce or manipulate it in any way.

To define the three strings, the user can modify the UNIX ASCII file `$APS_ETC/imgBrowse.opt`. This file may have the standard shell script comments (that is, lines that start with #). The strings are defined using the keywords (`string1`, `string2`, and `string3`) for the top, middle, and bottom strings. The equal sign (=) that separates the keyword and its value should have not spaces between the two. The string does not have to be in quotes. For example,

```
# My image browse options
string1=UNCLASSIFIED
string2=Approved for Public Release
string3=Distribution Unlimited
```

Selecting the browse image format

By default the APS system will make browse images using the PNG format. This format was selected over the JPEG format since the compression of that format leads to “halos” around the grid lines of the image. The JPEG compression was designed for natural photographs and not line-art.

The command **imgBrowse** may produce several image outputs besides PNG or JPEG. It can also produce PNM, TIFF, RGB, KML, EPS, PS, PDF and SVG. The APS top <format> node that controls this output.

For example, to produce GeoTiFF files (the images will then contain navigation), set <format> element to `tiff`.

```
<l3_browse>
    <format>tiff</format>
</l3_browse>
```

For example, to produce JPEG file with an associated ESRI World File (which will allow navigation of this image), set <format> to `tiff` and the `-U` in the <options>.

```
<l3_browse>
    <format>tiff</format>
    <options>-U</options>
</l3_browse>
```

Building a true color image from components

By default the APS system will make true color browse images using the surface reflectance products. The **imgBrowse** may generate an RGB image given three products separated by colons. In normal usage, this product has no landmask or flags overlaid on it. Therefore, we set the `<product>` to the three input bands and the appropriate **imgBrowse** options in `<options>`.

```
<l3_browse>
    <image>
        <product>rhos_660:rhos_555:rhos_443</product>
        <options>-T long_name=True_Color -M draw=0 -l draw=0</options>
        <alias>true_color</alias>
    </image>
</l3_browse>
```

The `<alias>true_color</alias>` is used to set the product in the output filename.

Data Retrieval Through Search

The APS driver has a thread that handles transfers of data to and from the system. To search for data from another archive like LAADS, LANCE, CLASS, etc., the user should provide an XML configuration file that provides parameters used to login to the remote host to identify the files to obtain.

For each file that was identified, an entry is made into the transfer data base indicating information required to obtain that file. Using this database, the APS driver will execute a download thread (or threads depending upon configuration). This download thread will obtain the file and mark the entry obtained when complete.

Through the database, the APS driver's downloader can be robust. It can handle issues when the hosts disk drives are full or offline; the network is down; or the remote host is down. It can handle partial downloads starting from where it left off.

Below is an XML file used to obtain a VIIR CLASS subscription for VIIRS EDR OCC file. This file must be located at `$APS_ETC/downloads.xml` by default. However, the path is configurable.

```
<downloads>

     <!-- VIIRS EDR AAOT CLASS Subscription (#6834) -->

     <download name="noaa-class-vocco-aaot" active="yes" process="no">
         <host>ftp.class.ncdc.noaa.gov</host>
         <chdir>sub/username/6834</chdir>
         <listing>*npp_d%Y%M%D*</listing>
         <sqlregex>*npp_d%Y%M%D*</sqlregex>
         <!-- file.sub(/^\d+[.]/,"") -->
         <loc_name>/^\d+[.]/,""</loc_name>
         <back>3</back>
     </download>

</downloads>
```

To activate this automatic download the primary APS configuration file must be set to turn on the downloads and specify the database used to handle the automated download of APS. See the aps man page for detailed information on the configuration file.

The listing below is sufficient to turn on the above downloads:

```
<transfer>
     <threads>1</threads>
     <db>pg:host:database</db>
</transfer>
```

Examining the remote sensing data base

The remote sensing data base has many SQL tables for maintaining information about the data files (and images) processed by APS. The `rsdb_main` is the primary data base which has a one to one relationship with any file known to the APS. In this section, some SQL examples are presented to control and query the remote sensing database.

The command **apsRSDB.rb** provides an interface to these tables. With this command, new files can be added to the database, registered files can be removed, and reports can be generated.

To find all GOCI Level-1 files that have no corresponding GOCI Level-2 file.

```
SELECT concat(p.path,'/',m.file) AS path
  FROM rsdb_main m, rsdb_paths p
 WHERE p.path_id = m.path_id
   AND m.main_id NOT IN (SELECT DISTINCT lvl1_id FROM rsdb_main m, rsdb_nrll2
 n WHERE m.main_id = n.main_id)
   AND m.ftype = 55
 ORDER BY file;
```

Find all AFWA files that cover the Gomex projection.

```
SELECT concat(p.path,'/',m.file) AS path, c.main_id, c.aoi_id, a.name,
 c.coverage, c.isp, c.iep, c.isl, c.iel
  FROM rsdb_coverage c, rsdb_aois a, rsdb_main m, rsdb_paths p
 WHERE a.aoi_id = c.aoi_id AND a.name = 'Gomex' AND c.coverage > 0 AND
 m.main_id = c.main_id
   AND m.ftype = 90
   AND m.path_id = p.path_id
   AND file like '%afwa_ops.h5'
 ORDER BY path;
```

Find all VOCCO files that cover MOBY for April 27, 2014 to May 20, 2014

```
SELECT m.file, c.main_id, a.name, c.coverage AS c, c.isp, c.iep, c.isl, c.iel,
 t.sdatetime
```

```
  FROM rsdb_coverage c, rsdb_aois a, rsdb_main m, rsdb_paths p,
 rsdb_timeframes t
 WHERE t.timeframe_id = m.timeframe_id
   AND a.aoi_id = c.aoi_id AND a.name = 'MOBY' AND c.coverage > 0 AND
 m.main_id = c.main_id
   AND m.ftype = 97
   AND m.path_id = p.path_id
   AND t.sdatetime > '2014-04-27T00:00:00' AND t.sdatetime <
 '2014-05-20T23:59:59'
 ORDER BY file;

 path                       | c | isp  | iep  | isl  | iel  | sdatetime
 VO...d20140427_t00193...h5 | 4 |  418 |  698 |  908 | 1219 | 2014-04-27T00:1
 VO...d20140428_t00020...h5 | 4 |  957 | 1199 |  185 |  492 | 2014-04-28T00:0
 VO...d20140428_t23385...h5 | 4 | 1491 | 1800 | 2547 | 2860 | 2014-04-28T23:3
 VO...d20140429_t23212...h5 | 4 | 2069 | 2329 | 1840 | 2160 | 2014-04-29T23:2
 VO...d20140430_t23035...h5 | 4 | 2544 | 2863 | 1163 | 1495 | 2014-04-30T23:0
 VO...d20140501_t00462...h5 | 2 |    1 |   36 |  168 |  465 | 2014-05-01T00:4
 VO...d20140501_t22462...h5 | 2 | 3068 | 3200 |  509 |  864 | 2014-05-01T22:4
 VO...d20140502_t00231...h5 | 4 |  242 |  507 | 2474 | 2785 | 2014-05-02T00:2
 VO...d20140512_t22360...h5 | 4 | 3197 | 3200 | 3064 | 3072 | 2014-05-12T22:3
 VO...d20140512_t22414...h5 | 4 | 3178 | 3200 |    1 |  146 | 2014-05-12T22:4
 VO...d20140513_t00182...h5 | 4 |  416 |  697 | 1800 | 2104 | 2014-05-13T00:1
 VO...d20140514_t00005...h5 | 4 |  956 | 1199 | 1057 | 1364 | 2014-05-14T00:0
 VO...d20140514_t23432...h5 | 4 | 1490 | 1800 |  343 |  656 | 2014-05-14T23:4
 VO...d20140515_t23201...h5 | 4 | 2068 | 2329 | 2721 | 3041 | 2014-05-15T23:2
 VO...d20140516_t23024...h5 | 4 | 2544 | 2863 | 2043 | 2374 | 2014-05-16T23:0
 VO...d20140517_t00451...h5 | 2 |    1 |   36 | 1048 | 1345 | 2014-05-17T00:4
 VO...d20140517_t22451...h5 | 2 | 3068 | 3200 | 1387 | 1733 | 2014-05-17T22:4
 VO...d20140518_t00274...h5 | 4 |  242 |  506 |  280 |  584 | 2014-05-18T00:2
 VO...d20140519_t00042...h5 | 4 |  780 | 1038 | 2617 | 2923 | 2014-05-19T00:0
 VO...d20140519_t23465...h5 | 4 | 1294 | 1594 | 1883 | 2194 | 2014-05-19T23:4
 VO...d20140520_t23292...h5 | 4 | 1894 | 2152 | 1181 | 1498 | 2014-05-20T23:2
```

Find GOMEX Level-3 files that make up the 7-day latest pixel composite for May 15th to May 22nd, 2013

```
SELECT concat(p.path,'/',m.file) AS path, m.main_id, t.sdatetime
  FROM rsdb_nrll3 n, rsdb_aois a, rsdb_main m, rsdb_paths p, rsdb_timeframes
 t
 WHERE t.timeframe_id = m.timeframe_id
   AND m.path_id = p.path_id
   AND a.aoi_id = n.aoi_id AND a.name = 'Gomex' AND n.main_id = m.main_id
   AND t.sdatetime > '2013-05-15T00:00:00'
   AND t.sdatetime < '2013-05-22T23:59:59'
   AND file LIKE '%L3_OC%'
 ORDER BY path;
 | path                                     |   main | sdatetime           |
 | /rs...3/135/npp.2013135.0515.171959...hdf | 436385 | 2013-05-15T17:19:59 |
 | /rs...3/135/npp.2013135.0515.185643...hdf | 436477 | 2013-05-15T18:56:43 |
 | /rs...3/135/npp.2013135.0515.190224...hdf | 436686 | 2013-05-15T19:02:24 |
 | /rs...3/136/npp.2013136.0516.201557...hdf | 436420 | 2013-05-16T20:15:57 |
 | /rs...3/137/npp.2013137.0517.181602...hdf | 436573 | 2013-05-17T18:16:02 |
```

```
| /rs...3/137/npp.2013137.0517.182144...hdf | 448986 | 2013-05-17T18:21:44 |
| /rs...3/137/npp.2013137.0517.195828...hdf | 436497 | 2013-05-17T19:58:28 |
| /rs...3/137/npp.2013137.0517.200408...hdf | 449005 | 2013-05-17T20:04:08 |
| /rs...3/138/npp.2013138.0518.175832...hdf | 448981 | 2013-05-18T17:58:32 |
| /rs...3/138/npp.2013138.0518.180414...hdf | 436974 | 2013-05-18T18:04:14 |
| /rs...3/138/npp.2013138.0518.194058...hdf | 448982 | 2013-05-18T19:40:58 |
| /rs...3/138/npp.2013138.0518.194639...hdf | 447559 | 2013-05-18T19:46:39 |
| /rs...3/139/npp.2013139.0519.174102...hdf | 435797 | 2013-05-19T17:41:02 |
| /rs...3/139/npp.2013139.0519.174643...hdf | 437136 | 2013-05-19T17:46:43 |
| /rs...3/139/npp.2013139.0519.192327...hdf | 437002 | 2013-05-19T19:23:27 |
| /rs...3/140/npp.2013140.0520.172332...hdf | 438190 | 2013-05-20T17:23:32 |
| /rs...3/140/npp.2013140.0520.190016...hdf | 436510 | 2013-05-20T19:00:16 |
| /rs...3/140/npp.2013140.0520.190557...hdf | 436010 | 2013-05-20T19:05:57 |
| /rs...3/141/npp.2013141.0521.184245...hdf | 436285 | 2013-05-21T18:42:45 |
| /rs...3/141/npp.2013141.0521.184827...hdf | 436022 | 2013-05-21T18:48:27 |
```

Find all AFWA files that have not been tested for AAOT coverage

```
SELECT concat(p.path,'/',m.file) AS path
  FROM rsdb_main m, rsdb_paths p
 WHERE p.path_id = m.path_id
   AND m.file LIKE '%afwa_ops.h5'
   AND m.main_id NOT IN (select main_id from rsdb_coverage where aoi_id = 4);
```

Find all Level-3 files which were generated from a Level-2 file which covered the Gulf Of Mexico projection and was processed using version 1 algorithms.

```
SELECT concat(p.path,'/',m.file) AS path
  FROM rsdb_main m, rsdb_paths p, rsdb_nrll3_inputs o
 WHERE p.path_id = m.path_id AND m.main_id = o.main_id
   AND o.lvl2_id IN (SELECT m.main_id FROM rsdb_coverage c, rsdb_aois a,
 rsdb_main m, rsdb_paths p, rsdb_nrll2 n
                     WHERE a.aoi_id = c.aoi_id AND a.name = 'Gomex' AND
 c.coverage > 0 AND m.main_id = c.main_id
                       AND m.ftype = 78 AND m.path_id = p.path_id AND
 m.main_id = n.main_id and n.proc_ver = 1 ORDER BY path);
```

Find the top twenty most cloud free GOCI Level-3 'Ocean Color' suite files.

```
 SELECT concat(p.path,'/',m.file) AS path, n_pixels-LAND AS npixls, CLDICE,
 100.0*CLDICE/(n_pixels-LAND) AS perc
  FROM rsdb_main m, rsdb_paths p, rsdb_flags f, rsdb_sensors s
 WHERE m.main_id = f.main_id AND m.path_id = p.path_id
   AND m.sensor_id = s.sensor_id AND s.sname = 'goci'
   AND m.file LIKE '%L3_OC%'
ORDER BY f.CLDICE LIMIT 20;
+-----------------------------------------------------+--------+--------+-----
| path                                                | npixls | CLDICE | perc
```

```
+------------------------------------------------------+--------+--------+-----
| /rs...do/2013/365/coms.2013365.1231.041537...0m.nc | 102400 |      0 | 0.00
| /rs...do/2013/365/coms.2013365.1231.031537...0m.nc | 102400 |      0 | 0.00
| /rs...do/2013/285/coms.2013285.1012.021535...0m.nc | 102400 |      0 | 0.00
| /rs...do/2013/285/coms.2013285.1012.031535...0m.nc | 102400 |      0 | 0.00
| /rs...do/2013/285/coms.2013285.1012.041535...0m.nc | 102400 |      0 | 0.00
| /rs...do/2013/194/coms.2013194.0713.021537...0m.nc | 102400 |      0 | 0.00
| /rs...do/2013/194/coms.2013194.0713.041537...0m.nc | 102400 |      0 | 0.00
| /rs...do/2013/194/coms.2013194.0713.031537...0m.nc | 102400 |      0 | 0.00
| /rs...do/2011/149/coms.2011149.0529.051536...0m.nc | 102400 |      0 | 0.00
| /rs...do/2014/081/coms.2014081.0322.041538...0m.nc | 102400 |      0 | 0.00
| /rs...do/2011/102/coms.2011102.0412.011538...0m.nc | 102400 |      0 | 0.00
| /rs...do/2014/074/coms.2014074.0315.031535...0m.nc | 102400 |      0 | 0.00
| /rs...do/2012/118/coms.2012118.0427.021537...0m.nc | 102400 |      0 | 0.00
| /rs...do/2012/118/coms.2012118.0427.041536...0m.nc | 102400 |      0 | 0.00
| /rs...do/2012/118/coms.2012118.0427.031536...0m.nc | 102400 |      0 | 0.00
| /rs...do/2012/138/coms.2012138.0517.031537...0m.nc | 102400 |      0 | 0.00
| /rs...do/2012/138/coms.2012138.0517.041537...0m.nc | 102400 |      0 | 0.00
| /rs...do/2012/085/coms.2012085.0325.041537...0m.nc | 102400 |      0 | 0.00
| /rs...do/2012/085/coms.2012085.0325.031538...0m.nc | 102400 |      0 | 0.00
| /rs...do/2012/074/coms.2012074.0314.031537...0m.nc | 102400 |      0 | 0.00
+------------------------------------------------------+--------+--------+-----
20 rows in set (2.16 sec)
```

Examining the transfer data base

With the transfer data base SQL tables `downloads` and `uploads`, APS can automatically handle downloading (or uploading) data to/from APS. In this section, some SQL examples are presented to control and query these transfers.

The command **apsFileTransferDB.rb** provides an interface to these tables. With this command, a new upload or download entry can be added to the tables. Additionally, this command provides for report generation.

For example, to obtain the ancillary data file `N201236600_SEAICE_NSIDC_24h.hdf` from Goddard.

```
$ f=N201236600_SEAICE_NSIDC_24h.hdf
$ apsFileTransferDB.rb --download http://oceandata.sci.gsfc.nasa.gov/cgi/
getfile/$f /home/aps/aps_v6.4.4/work/$f
```

In this example, we obtain all the ozone ancillary data for the days 300 to 309 of the year 2012. Again this data is obtained from Goddard.

```
 for d in 300 301 302 303 304 305 306 307 308 309;
 do
     f=S2012${d}00${d}23_TOAST.OZONE
     echo $f
     apsFileTransferDB.rb --download http://oceandata.sci.gsfc.nasa.gov/cgi/
getfile/$f /home/aps/aps_v6.4.4/work/$f
 done
```

One might access the SQL data base directly using a tool provided by the data base vendor. For example, if the transfer data base resides on `host1` running MySQL, then the user might use **mysql**. If using PostgreSQL, then the command would be **psql**. Below are examples of using this command.

```
$ mysql --password=password -h host1 rsdb
Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql>
```

```
$ psql -h host1 -U aps rsdb
Password for user aps:
psql (8.4.20, server 8.1.23)
WARNING: psql version 8.4, server version 8.1.
         Some psql features might not work.
         Type "help" for help.

rsdb=>
```

To see which files are `IN_PROGRESS`, we can check for status entries of -1.

```
SELECT * FROM downloads WHERE status = -1;
```

To reset a particular download to force it to download the file again.

```
UPDATE downloads SET attempts=0, status=0, done=0, finished=0, start=queued
WHERE remote_file = 'filename';
```

To force a download that was set to begin acquistion at some time in the future to delay another hour , we reset its start time by 360 seconds.

```
UPDATE downloads SET start=start+360 WHERE remote_file = 'filename';
```

To determine the number of files that we aquired that where over 1.2GB in size. The second query will show them with their file size in order.

```
SELECT count(*) FROM downloads WHERE aquired > 1200000000;
+----------+
| count(*) |
+----------+
|     1508 |
+----------+
1 row in set (0.03 sec)

SELECT aquired, remote_file FROM downloads WHERE aquired > 1200000000 ORDER
BY aquired DESC LIMIT 5;
+------------+------------------------------------------------------------+
| aquired    | remote_file                                                |
+------------+------------------------------------------------------------+
| 2147483647 | MER_FRS_1PNUPA20110607_093055...3103_00180_48470_6775.N1.bz2 |
```

```
| 2147483647 | MER_FRS_1PNUPA20110608_085159...3103_00194_48484_6831.N1.bz2 |
| 2147483647 | MER_FRS_1PNUPA20110609_100637...3103_00209_48499_6882.N1.bz2 |
| 2147483647 | MER_FRS_1PNUPA20110609_113341...3103_00210_48500_6891.N1.bz2 |
| 1978309748 | MER_FRS_1PNUPA20110609_234242...3103_00217_48507_6908.N1.bz2 |
+------------+--------------------------------------------------------------+
5 rows in set (0.03 sec)
```

Part II. Data Format - Version 3.3

The sections in Part II contain a data user's guide which describes the files created by the Automated Processing System

The Regional Data Products produced by the Navy's Automated Processing System (aps) contains atmospherically corrected geophysical products in a standard map projection for a specific region of interest derived from one of several different satellites (GOCI, MERIS, MODIS, SeaWiFS and VIIRS). This Data Product Guide will describe version 3.4 of the aps file format.

Table of Contents

Chapter 5. The APS Format

The APS format is either a Hierarchical Data Format Version 5 - as realized by the netCDF v4 application programming interface (API) - or a Hierarchical Data Format Version 4 - as realized by the Scientific Data Sets API. In either case, the only objects used are data set objects and attributes.

The APS IO library contains routines for accessing all objects from the APS file. The use of the library is encouraged so that the underlying file structure may change. This document will describe the file format structure (that is, the names used for attributes and data sets) as well as the use of the library to get or obtain that particular object.

Standard APS Global Attributes

The following global attributes are required in all APS files. In the tables below, the data type of the attribute precedes the name of the attribute. If the attribute is an array (limited to a single dimension), the size is noted.

Table 5.1. File Attributes

Name	Description
file	The name of the product
fileTitle	see Table 5.2, “FileType Values”
fileVersion	4.0
createAgency	The agency which created the file
createSoftware	The version of the software which created the file
createPlatform	The hardware/software platform the file was created on
createTime	The date and time when the file was created
processedVersion	Processing version number
orbitNumber	Satellite orbit number
along-track	Number of lines
cross-track	Number of pixels

The fileType attribute describes the type of data stored in the APS file. It may contain any of the following values.

Table 5.2. FileType Values

Value	Description
NRL Level-2	NRL file containing Level-2 data
NRL Level-3	NRL file containing Level-3 data
NRL Level-3 Mosaic	NRL file containing a mosaic of several Level-3 data files
NRL Level-4	NRL file containing Level-4 data

This information provides information about the starting and ending time of the data used to create this product. This information is relative to the input product file.

Table 5.3. Time Attributes

Name	Description
timeStart	UTC start time as an ASCII string

Name	Description
timeStartYear	UTC year of data start, e.g. 2003
timeStartDay	UTC day-of-year of data start (1-366)
timeStartTime	UTC milliseconds-of-day of data start (1-86400000)
timeEnd	UTC end time as an ASCII string
timeEndYear	UTC year of data end, e.g. 2003
timeEndDay	UTC day-of-year of data end (1-366)
timeEndTime	UTC milliseconds-of-day of data end (1-86400000)
timeDayNight	Flag indicating if data collected during day or night. May be one of Day, Night, Day/Night

These attributes are used to describe the specific sensor from which the Level-3/Level-4 Regional Data Product was derived. The APS software can handle several data streams including GOCI, MERIS, MODIS, SeaWiFS and VIIRS.

Table 5.4. Sensor Attributes

Name	Description
sensor	GOCI, MERIS, MODISA, SeaWiFS, VIIRSN
sensorPlatform	Platform carrying sensor, like Orbview-2, MODIS-AQUA, NPP, ENVISAT-1, COMS
sensorAgency	Agency/Owner of Sensor
sensorType	Type of sensor: scanner, pushbroom, whiskbroom
sensorSpectrum	Description of spectrum: visible, near-IR, thermal
sensorNumberOfBands	Number of Bands
sensorBandUnits	Units of wavelengths, like nm
sensorBands	Center wavelengths
sensorBandWidths	Nominal width of bands
sensorNominalAltitudeInKM	Nominal Altitude of sensor
sensorScanWidthInKM	Distance on earth of Field of View in kilometers
sensorResolutionInKM	Distance on earth of a single pixel in kilometers
sensorPlatformType	Type of platform

These attributes provide information about the number of times a particular flag was set. These flags come from the `l2_flags` product.

Table 5.5. Flag Count Attributes

Name	Description
flagCountATMFAIL	Atmospheric failure
flagCountLAND	Land pixels
flagCountPRODWARN	Product warning
flagCountHIGLINT	High glint
flagCountHILT	High radiance or missing data
flagCountHISATZEN	High satellite zenith

Name	Description
flagCountCOASTZ	Coastal
flagCountBLOTCH	Pixels in a blotch pattern
flagCountSTRAYLIGHT	Straylight contamination
flagCountCLDICE	Clouds or ice
flagCountCOCCOLITH	Pixels flagged as COCCOLITH
flagCountTURBIDW	Pixels flagged as turbid water
flagCountHISOLZEN	High solar zenith angle
flagCountTRIMPIXEL	VIIRS trimpixel data
flagCountLOWLW	Low Lw at 550
flagCountCHLFAIL	Chlorophyll-a failure
flagCountNAVWARN	Navigation warning
flagCountABSAER	Absorbing aerosol
flagCountSTUMPF	Stumpf 412 iteration
flagCountMAXAERITER	Maximum iteration limiit hit
flagCountMODGLINT	Moderate glint
flagCountCHLWARN	Chlorophyll-a warning
flagCountATMWARN	Atmospheric warning
flagCountSEAICE	Seaice presence
flagCountNAVFAIL	Navigation failure
flagCountFILTER	Data filtering failure
flagCountSSTWARN	SST warning
flagCountSSTFAIL	SST failure
flagCountHIPOL	High polarization
flagCountPRODFAIL	Product failure
flagCountOCEAN	Ocean pixels

NASA Global Attributes

The following global attributes are required to allow APS files to be read directly by the SeaDAS (v7 and above).

Table 5.6. NASA Attributes

Name	Description
title	Title of product
time_coverage_start	Start time
time_coverage_end	Edn time
start_center_longitude	NADIR longitude at start of pass
start_center_latitude	NADIR latitude at start of pass
end_center_longitude	NADIR longitude at end of pass
end_center_latitude	NADIR latitude at end of pass

Name	Description
northernmost_latitude	Northern most latitude
southernmost_latitude	Southern most latitude
easternmost_longitude	Eastern most longitude
westernmost_longitude	Western most longitude
geospatial_lat_units	Unit of latitudes
geospatial_lon_units	Unit of longitudes
geospatial_lat_max	Northern most latitudee
geospatial_lat_min	Southern most latitude of platform
geospatial_lon_max	Eastern most longitude
geospatial_lon_min	Western most longitude
startDirection	Ascending or Descending
endDirection	Ascending or Descending
day_night_flag	Day, Night, or Both

BEAM Global Attributes

The following global attributes are required allow APS files to be read by the BEAM software directly.

Table 5.7. BEAM Attributes

Name	Description
start_date	Start date
start_time	Start time
stop_date	Stop date
stop_time	Stop time

Chapter 6. Level 3 Regional Data Products

A Level-3 Regional Data Product file contains atmospherically corrected geophysical products in a standard map projection for a specific region of interest derived from one of several different satellites (GOCI, MERIS, MODIS, SeaWiFS, VIIRS). A Level-3 Regional Data Product can be stored in one of three underlying formats. The default format is the Hierarchical Data Format (HDF) developed by the National Center for Supercomputer Applications (NCSA) at U. of Illinois Urbana-Champaign (version 4.2.8). Additionally, version 5 of the HDF format can be written (using HDF5 v1.8.10) as well as the netCDF v3/v4 (using netCDF v4.2) format.

Naming Convention

The form of a Level-3 Regional Data Product file name is `satellite.yyyyddd.mmdd.X.L3.sensor.region.version.resolution.format`, where `satellite` is

- `aqua`
- `coms`
- `envisat-1`
- `orbview-2`
- `npp`

`yyyyddd.mmdd.hhmmss` is the concatenated digits for the UTC year, day of year, month of year, day of month, hours, minutes and seconds of the first scan line processed to form the Level-3 file. `X` is a day/night flag indicator:

- `D`for day-time only scans
- `N` for night-time only scans
- `B` for both day and night scans
- `U` for unknown

`sensor` is

- `goci` for GOCI
- `meris` for MERIS
- `modis` for Moderate Resolution
- `seawifs`for SeaWiFS
- `viirs` for VIIRS

`region` is a user-defined code to represent the region. `resolution` is a indication of the resolution of the original input data. `version` is the algorithm processing version. `format` may be one of:

- `hdf`for HDF v4 format
- h5 for HDF v5 format

- nc for netCDF

Examples of Level-3 Regional Data Product file names are:

- `aqua.2003030.0130.190000.D.L3_SST.modis.MSB.v06.1000m.hdf`
- `envisat-1.2008177.0625.183241.D.L3.meris.MNT.v03.300m.hdf`
- `coms.2013030.0130.021537.D.L3_QAA.goci.YellSea.v01.500m.hdf`
- `orbview-2.2001148.0528.181616.D.L3.seawifs.SWR.v07.1100m.hdf`

File Attributes

The file attributes are associated with all products in the HDF file. These attributes contain such information as the creation of the file, the sensor platform for which the data was derived, and geographical coverage. The attributes are divided into several groups and are discussed below.

The processing meta data includes a description of the inputs used to create this Level-3 Regional Data Product file. Not all attributes given here will be present in a given file - several are sensor specific.

The `prodList` attribute is a list of the geophysical products stored in the file. It excludes the standard data sets like CP_Pixels, etc.

Table 6.1. Input Parameters Attributes

Name	Description
inputMasksInt	The mask defined as an integer
inputMasks	A comma seperated list of flags that were used as masks during processing.
prodList	A comma seperated list of products stored in this file.
processingVersion	Version of processing

This meta data contains information about the geographical coverage of the data file and the resulting regional geographical coverage.

Because the Level-3 Regional Data Product files are mapped images, the following information provides the navigation information of the projection used. The projection system used by APS 6.4.4 is based on the NASA PC-SeaPAK projection system which uses two tie-points and the USGS projection software to perform the navigation.

Table 6.2. Navigation Attributes

Name	Description
navType	Navigation type of data. Always set to 'mapped'
mapProjectionSystem	Map projection system used. Always set to NRL(USGS)
mapProjection	Name of the SDS included in the file that contains the map projection parameter values.
mappedUpperLeft	Latitude and longitude of upper left (1,1) point of each product.
mappedUpperRight	Latitude and longitude of upper right (1,n) point of each product.
mappedLowerLeft	Latitude and longitude of lower left (m,1) point of each product.
mappedLowerRight	Latitude and longitude of lower right (m,n) point of each product.

Special Data Sets

There are a *variable* number of data sets in an APS Level-3 Regional Data Product file. However, several data sets are standard. These data sets are meta data sets containing information related to the actual geophysical data products. The first group of data sets listed below provide geographical coverage and data quality information. The last group of data sets are the actual geophysical products and vary in number.

The following data sets provide for a grid of control points that relate the image locations (pixel,line) with a geographical location (lat,lon). The grid is a 2-D regular “square” grid with each latitude and longitude positioned at the given pixel and line locations. These are single dimensional arrays indicating the pixel and line locations for each grid point. Though every Level\-3 Regional Data Product file is map projected, this control points array is provided to give the user an alternative method of georeferencing each pixel in the data products.

Table 6.3. Geographical Location Datasets

Map Projections	
Name	**Description**
mapProjection	Map projection parameters
Control Points	
Name	**Description**
CP_Pixels[m]	Control Point pixel locations
CP_Lines[n]	Control Point line locations
CP_Latitudes[m][n]	Latitude in decimal degrees for each control point
CP_Longitudes[m][n]	Longitude in decimal degrees for each control point

The data quality data sets provide information about the data quality of each pixel in the geophysical product data sets.

Table 6.4. Data Quality Datasets

Name	Description
l2_flags	Level-2 Processing Flags

The l2_flags contain 32 bits. Each bit represents a condition or state for the pixel in question. Each bit and their associated keywords are shown in the following table.

Table 6.5. l2_flags Bit Flag Values

Bit	Name	Description
1	ATMFAIL	Atmospheric correction algorithm failure
2	LAND	Land pixel
3	PRODWARN	Product warning
4	HIGLINT	Sun glint
5	HILT	Total radiance greater than the knee value
6	HISATZEN	Large sensor zenith angle
7	COASTZ	Shallow water (>30m deep)
8	BLOTCH	Pixel was used in a statistical region (see regions_file parameter)

Bit	Name	Description
9	STRAYLIGHT	Stray Light
10	CLDICE	Cloud or ice
11	COCCOLITH	Coccolithophores
12	TURBIDW	Case 2 waters
13	HISOLZEN	Large solar zenith angle
14	SPARE	unused
15	LOWLW	Low water-leaving radiance at band 5
16	CHLFAIL	Chlorophyll algorithm failure
17	NAVWARN	Questionable navigation
18	ABSAER	Absorbing aerosol
19	STUMPF	Absorbing aerosol correction (Stumpf 2007) applied to pixel
20	MAXAERITER	NIR algorithm exceeded maximum iteration
21	MODGLINT	Moderate Sun glint
22	CHLWARN	Chlorophylla out of range
23	ATMWARN	Epsilon out of range
24	SPARE	unused
25	SEAICE	Sea/ice
26	NAVFAIL	Navigation failure
27	FILTER	Data filtering failure
28	SSTWARN	SST out of range
29	SSTFAIL	SST computation failure
30	HIPOL	High polarization
31	PRODFAIL	Product algorithm failure
32	OCEAN	Ocean or "water" pixel (that is not LAND or CLDICE)

Product Data Sets

The remaining data sets in the Level-3 Regional Data Product data file are general in nature and may or may not exist in the given data set. A description of each product follows. All products below that exist in the Level-3 Regional Data Product file are listed in the file attribute `prodList`.

To describe the product, several attributes are attached to the data set. These include a long descriptive name, the units and valid range of the data.

The long descriptive name is an ASCII string. In some cases, the algorithm used is also provided in the name. Examples are “Remote Sensing Reflectance at 443 nm” and “Chlorophyll Concentration, OC4 Algorithm”.

The units are stored in an ASCII string. Examples of unit strings are “mg m^-3” and “mW cm^-2 um^-1 sr^-1”.

The valid range attribute provides a suggested range of valid data. This attribute can be used to filter out values that are outside the given range, for example, for compositing several scenes together. In general, values outside this range are considered suspect.

Additionally, the value used to represent invalid values is also defined. An invalid value represents locations where a value can not be computed. For example, when trying to compute a chlorophyll-a value in which the remote sensing reflectance at 443 nm is negative. Or, when a particular pixel is over land and the land is being mask out (the normal procedure).

In general, most of the archival products are stored as signed 16-bit integers. In all cases for products produced by APS to date, the scaling is linear. Note, that not all products are stored as integers; some are stored as floating point values. The products listed below that indicate floating point representation are generally not archival products. In the cases of floating point products, these attributes are not available.

If the product is stored as integers, then it will also have the next two attributes, which are the slope and intercept for converting the integers into floating points. The integer value should be read from the SDS and multiplied by the slope and then have the intercept added.

These attributes are for the “standard” units of the product. If the product has the attribute `additionalUnits` set, then it will have additional attributes similar to these that allow for the conversion of the data in those units. The attributes will be named according to the unit name. For example, if `additionalUnits` contains “Fahrenheit” the product's slope and intercept will be contained in the attributes `FahrenheitSlope` and `FahrenheitIntercept`. See the example below

To provide for making quick-look browse images, suggestions about the scaling and data ranges for making images of the product data are stored in the following two attributes. The APS program imgBrowse will use these attributes as default scaling and data ranges for automatic creation of quick-look browse images.

For some products, like diffuse attenuation, it is generally accepted that a logarithmic scaling for images brings out the best detail. Therefore, for this product a logarithmic function will be used. Other products like sea surface temperature are generally best displayed using linear scaling.

The display range of the data may or may not be the same as the `validRange` attribute above since in some cases (e.g. rrs_412), the data has been known to fall outside a geophysical valid range, but we wish to display the non physical data since some physical oceanographical structure or feature may still be present in the data.

Table 6.6. Product Data Set Attributes

Name	Description
productClass	This is a description of the classification of the product. Entries may be “geo-physical”, “image”, “flags”, etc.
productName	This is a description of the product.
productAlgorithm	This is a notation about the algorithm, usually a paper reference.
productUnits	This is a description of the units of the product.
additionalUnits	This is a space delimetered string of additional units available for this product. For example, an sst product may set this string to “Kelvin Fahrenheit”
productStatus	This new SDS attribute will give an indication of the status this product.
validRange	This is a suggested range of valid data.
badData	This is the geophysical value which will represent invalid data for the given product.
productScaling	The type of scaling of the product. Currently, always Linear
scalingSlope	The slope for product scaling.
scalingIntercept	The intercept for product scaling.

Name	**Description**
browseFunc	This is a suggested function to apply to convert the data in the SDS into an image. A value of 1 indicates linear scaling; a value of 2 indicates log10 scaling.
browseRanges	This is a suggested display range when converting the data in the SDS into an image. This may or may not be the same as validRange because in some cases (e.g. rrs_412), the data has been known to fall outside the range, but we wish to display the invalid data. This attribute is used by the APS program imgBrowse when creating quick-look browse images of different products.
browseColorTableName	This is a suggested color table to use for image. The names correspond to those used by **imgBrowse**.

Chapter 7. Level-4 Regional Data Products

A Level-4 Regional Data Product file contains atmospherically corrected geophysical products in a standard map projection for a specific region of interest derived from one of several different satellites (GOCI, MERIS, MODIS, SeaWiFS, VIIRS). A Level-4 Regional Data Product may be stored in one of several formats. The default is stored using the Hierarchical Data Format (HDF) developed by the National Center for Supercomputer Applications (NCSA) at U. of Illinois Urbana-Champaign (version 4.2.8). Additionally, version 5 of the HDF format can be written (using HDF5 v1.8.10) as well as the netCDF v3 format.

File Attributes

The file attributes are associated with all products in the HDF file. These attributes contain such information as the creation of the file, the sensor platform for which the data was derived, and geographical coverage. The attributes are divided into several groups and are discussed in the the section called “File Attributes”

The Level-4 Regional Composite Data Product Files also contain an additional set of attributes.

Table 7.1. Input Parameters Attributes

Name	**Description**
compType	Set to `Daily Composite', `Weekly Composite', `Monthly Composite', `Yearly Composite', `Rolling Composite`, `Latest Pixel Composite', `Seasonal Composite'
compStartTimeFrame	The intended start time of the composite (not set for `Rolling Composite' or `LatestPixel Composite'
compEndTimeFrame	The intended end time of the composite (not set for `Rolling Composite' or `LatestPixel Composite'
compTimeFrame	The time frame of composite in days (like 8, 30, 31, 365)
compMaxPixels	The maximum number of pixels used in the latest pixel composite. Required only for `Latest Pixel Composite'
inputFiles	A list of files used to produce the average product

Special Data Sets

There are a *variable* number of data sets in an APS Level-4 Regional Data Product file most of which contain a oceanographic parameter. However, several data sets are special. These data sets are meta data sets containing information related to the actual geophysical data products. These data sets are described in the section called “Special Data Sets”.

Product Data Sets

The product data sets of a Level-4 Regional Data Product file are similar to Level-3 Regional Data Products described in the section called “Product Data Sets”. The difference in the Level-4 files is that the product is an average of the input files.

Additionally, other data sets which describe the minimum, maximum, standard deviation, and number of pixels may also be present. These data sets use the same name as those described in above, except that an `_min`, `_max`,`_stddev`, and `_num` are appended to each product name. For example, the file may contain the data set `sst_max`, which is the maximum composite of the `sst` product.

Part III. Ocean Color Satellites

Table of Contents

Chapter 8. Ocean Color Processing

The ocean color processing used in aps is based on the SeaDAS processing software package **l2gen**. However, the aps makes several changes to the the NASA/Goddard system. In summary, these include: (1) ability to run hyper-spectral data (NBANDS=128), (2) several methods for land detection, (3) the use of the Stumpf 412 reflectance iteration, (4) several additional products including water mass classification products, hypoxia products, and visibility products, and (5) ability to output an APS file (which can be HDFv4, HDFv5, or netCDF-base).

Land Detection

When set to “0”, the default SeaDAS-based land mask and API routine is used (`$APS_DATA/common/landmask.dat`). When set to “2”, the APS API Aps_LandMask is used. This routine can handle up to four different types of land mask files. The first is the SeaDAS-based land mask. The second and third files are user generated land masks. The second land mask is a NSIPS (Naval Satellite Image Processing System) land mask. The third land mask is an HDF-based user defined land mask in an APS format with APS mapping information. The fourth file is the 250 m land mask (`$APS_DATA/common/MOD44w.h5`). This land mask was generated from the MODIS 44w product.

When set to “2”, the `land_Lt` and `land_ndvi` variables are used to control a NDVI-based land detection. This method is highly sensitive to scene in question.

See FIXME for examples of a 300m MERIS file of the Mississippi River Delta showing the use of the original SeaDAS-based land mask and the 250m MOD44w land mask.

Stumpf 412 Iteration

The APS uses the reflectance iteration of Stumpf. In this method, an estimate is made to determine the expected reflectance at 412. If the estimate is higher than the atmospherically computed value, a power-law function (modeling an absorbing aerosol) is used to adjust the reflectance spectrum. In APS, this adjustment is propagated to the water-leaving radiance. See xxx for more discussion.

New products

The APS system includes several algorithms that are unique to the APS community. These include products like diver visibility (horizontal and vertical) and diffuse attenuation provided as a "length" in meters. Additionally, a beam attenuation estimate using the high-resolution (250 m) red-band (645 nm) is available.

The APS includes a group of products which collectively known as water mass classification. These include a division of the in-water absorption into its phytoplankton, sedimentary, yellow substance, and particulate components. From three of these properities (), a colored image based on these three axes is formed. This image provides a classification of the water mass.

APS output

The original **MSl12** program had the ability to output several different HDF (v4) formats. The APS software maintains that original capability that was removed in the latest version of (the now named) **l2gen** program. The `ofmt` is retained and used to control that output.

When `ofmt=3`, the APS format is selected. The APS format is (by default) an HDF v4 formatted file with a different set of global and data set attributes. It also includes additional data sets which are use by the APS **imgMap** warper.

In APS, also has the ability to automatically name the output based on information gathered from the data. For example, the standard APS file nomenclature includes a flag indicating if the file contains all night (N) pixels, all day (D) pixels, or a mixed set of day and night (B) pixels.

Cloud Detection

The standard NASA cloud test is a threshold-based quasi-surface near-infrared (865) reflectance based test. [http://oceancolor.gsfc.nasa.gov/REPROCESSING/SeaWiFS/R4/cloud/cloud_mask_ocean.html] The parameter `albedo` is used to control the threshold of the test. It defaults to 0.027. The parameter `cloud_wave` is used to control the NIR band wavelength on a per-sensor basis. To stop the further processing of pixels identified as cloud, the `maskcloud` parameter should be set to 1 (the default).

In addition to the standard NASA albedo theshold, **n2gen** provides the reflectance ratio test. This test is part of the MODIS ATBD MOD35: “Discriminating Clear-Sky From Cloud With MODIS”. For MODIS 250m processing, it allows for full-resolution cloud detection since the ratio test using the 1KM data.

The reflectance ratio test uses the 870 band divided by 660 band (R0.87/R0.66). This test makes use of the fact that the spectral reflectance at these two wavelengths is similar over clouds (ratio is near 1) and different over water and vegetation. Using AVHRR data this ratio has been found to be between 0.9 and 1.1 in cloudy regions. If the ratio falls within this range, cloud is indicated.

To control cloud detection algorithm selection, **n2gen** provides the `cloud_opt` parameter. A value of 0 (the default) selects the standard NASA algorithm and a value of 1 selects the NRL reflectance ratio test.

Chapter 9. Level-2 Flags

There are a series of bits that indicate various conditions or states of the data. These flags provide quality information about each pixel in the data set.

The ATMFAIL (bit 1) flag is used to indicate pixels for which the atmospheric correction failed.

The LAND (bit 2) flag is used to indicate pixels which contain land. There are several methods to use for setting this bit. The method is controlled by the `land_opt` option.

The PRODWARN (bit 3) flag is used to indicate pixels that have products that may not be optimal.

The HIGLINT (bit 4) flag is set only for non-land pixels and day time pixels . It is set whenever, the normalized glint radiance is above a given threshold. The input parameter `glint_thresh` is used to control this value. The default is 0.005.

The HILT (bit 5) flag is used to indicate pixels that were saturated at the sensor level.

The HISATZEN (bit 6) flag is used to indicate pixels whose satellite zenith angle is larger than a given threshold. The input parameter `satzen` is used to control this value. The default is 60.0.

The COASTZ (bit 7) flag is used to indicate pixels which are determined to be in water less than 30m. This determination is based on the water mask file `$APS_DATA/common/watermask.dat`..

Bit 8 is an unused flag or mask.

The STRAYLIGHT (bit 9) flag gives an indication of the stray light contamination of the given pixel.

The CLDICE (bit 10) flag is used to indicate that the pixel is either ice or cloud covered.

The COCCOLITH (bit 11) flag is set to indicate the presence of coccolithophores.

The TURBIDW (bit 12) flag is set when the remote sensing reflectance at 670 nm is larger than 0.012. This flag is not set if the Wang/Shi switching algorithm is used.

The HISOLZEN (bit 13) flag is used to indicate pixels for which the solar zenith angle is larger than a given threshold. The input parameter `sunzen` is used to control this value. The default is 75.0.

Bit 14 is an unused flag or mask.

The LOWLW (bit 15) flag is set when the normalized water leaving radiance at the green band (555 nm) is low or negative. The input parameter `nlwmin` is used to control this value. The default is 0.15

The CHLFAIL (bit 16) flag indicates that the computation for chlorophlly-a has failed.

The NAVWARN (bit 17) flag is set when the error in the navigation is large.

The ABSAER (bit 18) flag indicates that the pixel is contaminated by an absorbing aerosol.

The STUMPF (bit 19) flag is set when the Stumpf 412 iteration has modified the originally derived reflectance.

Note

In previous versions of APS, this was a different bit

The MAXAERITER (bit 20) flag indicates that the NIR iteration reached the maximum number of iterations and stopped.

The MODGLINT (bit 21) flag indicates that the pixel in question has moderate glint contamination. This bit is set on when the normalized glint radiance is greater than 0.0001.

The CHLWARN (bit 22) flag indicates that the chlorophyll-a concentration in not optimal.

The ATMWARN (bit 23) flag that the atmospheric correction for the given pixel is sub-optimal.

Bit 24 is an unused flag or mask.

The SEAICE (bit 25) flag indicates that the pixel has a high probability of containing sea ice. This is based an a sea ice climatology file. The input parameter `icefile` is used to control this value. The default is `$APS_DATA/common/ice_mask.hdf`

The NAVFAIL (bit 26) flag indicates that the navigation of the given pixel is invalid.

The FILTER (bit 27) flag is set when a filter is used and fails to provide valid data.

The SSTWARN (bit 28) flag indicates that the sea surface temperature has suboptimal quality. In general, this means that the qual_sst value is larger than 0.

The SSTFAIL (bit 29) flag indicates that the sea surface temperature is invalid. In general, this means that the qual_sst indicator is 4.

The HIPOL (bit 30) flag is set when the degree of polarization is higher than a given threshold. This threshold is controlled by `hipol` parameter. The default value is `0.5`.

The PRODFAIL (bit 31) flag indicates that the products generated for this pixel are suboptimal.

The OCEAN (bit 32) flag is used to indicate pixels that are water pixels. That is, not-land or cloud.

Part IV. Command Line Reference

The sections in Part II contain the reference pages for each program available in the Automated Processing System.

Name

aps — aps driver script

Synopsis

`aps.rb` setup [*`file`*]

`aps.rb` config [*`verbose`*]

`aps.rb` [--auto **|** --threads *n*] init

`aps.rb` [start **|** stop **|** kill **|** term]

Description

The **aps.rb** script is the driver for the Automated Processing System. The **aps.rb** is a Ruby script for a cluster of single or multiple processor systems (parallel processing). It is also the administrative interface to the Automated Processing System - the script the user controls the system.

install

The *install* command is used to install any Ruby Gems required to run APS. It should be executed once upon unpacking of the tar files.

setup

The *setup* command is used open a GUI interface for the creation of the configuration file to be used by the *init* command.

config [verbose]

The *config* command is used print the name of the configuration file. If no configuration file is found, nothing is returned.

The **aps.rb** uses an XML configuration file. The search pattern for this file has several different levels. To begin, if the environment variable `APS_CONFIG` is defined, that is taken to be the XML configuration file. Otherwise, the directory search pattern: `APS_ETC`, `APS_DIR/RUBY_PLATFORM/etc`, `APS_DIR/etc`, *`directory_of_executable`*`/etc`. In each directory, a host specific configuration file `aps.$(hostname).conf.xml` is looked for and then a standard configuration file `aps.conf.xml`.

If the optional keyword **verbose** is added to **config**, then a print out of all the parameters will occur. This can be used to verify the configuration settings prior to starting APS.

init

The *init* command is used to start up the APS driver. The prompt will return immediately, but the script has restarted itself in the background and will enter an infinite loop.

When initialized, the APS driver will create three files: `aps.$(hostname).pid`, `aps.$(hostname).process`, and `aps.$(hostname).log`. The first file contains the processing ID of the driver (**aps.rb**). It is used to terminate and/or kill a running APS driver. The second file is used to tell the APS driver whether to process any data. It is checked for existance once a minute (by default). If non-existant, no data will be processed. It provides the user a method to temporarily halt data processing without having to terminate the APS driver (see the *start* and *stop* commands).

The last file is a logging file. The APS driver will write messages about the files it has processed to this file. The APS log file is a “rolling” log file. The APS automatically appends a six digit sequential number preceded by a dash before the `.log` extension. For example, `aps.localhost.domain.net-000001.log`. The log files are “rolled” when the size reaches 256 MB.

With the **--auto** option (with command **init**), the **aps.rb** will automatically create all missing directories upon start up. Otherwise, if the directory structure is not correct, the Ruby controller will exit.

The **--threads** option for the **init** command, is used to define the number of threads. This option overrides any configuration values.

term

The *term* command is used to terminate the APS driver whose PID is in the file `aps.$(hostname).pid` by sending the process a TERM signal. When the TERM signal is caught, APS will issue TERM signals to all sub-processes.

kill

The *kill* command is used to immediately kill the APS driver whose PID is in the file `aps.$(hostname).pid` by sending the process a KILL signal. In this case, the driver will not send any signals to its sub-processs, so that, some processing may continue.

start

The *start* command creates the `aps.$(hostname).process` file which is used to signal the APS driver to proceed with data processing. Usually this is only called if *stop* had been previously run.

stop

The *stop* command deletes the `aps.$(hostname).process` file which is used to signal the APS driver to stop processing data. When the file is non-existant the APS driver will continue to poll the `in` directory, but will never process the data.

One use of the two commands *start* and *stop* is within *cron* to prevent the APS from running (and taking up computer resources) during the work day. This was its original intent, though the APS is now usually run on a dedicated system.

Environmental Variables

The environmental variables in this section are required to define the directory structure used by the APS system. Many of the environmental variables have defaults if the normal directory structure is followed. They should only be re-defined if your configuration requires different values. They are defined in the configuration file (`aps.conf.xml`) located in the `etc` directory. Modify the configuration file if you need to set these to some other values.

`APS_DIR` The top level directory for the APS system.

`APS_ETC` The location of configuration files for the APS system. Defaults to `$APS_DIR/etc`.

`APS_BIN` The location of executables for the APS system. Defaults to `$APS_DIR/bin` or `$APS_DIR/$target_cpu-$target_os/bin`.

`APS_LIB` The location of libraries for the APS system. Defaults to `$APS_DIR/lib` or `$APS_DIR/$target_cpu-$target_os/lib`.

APS_WORK	The directory where all the data processing will be performed. Defaults to $APS_DIR/work.
APS_OUT	The directory where all the final products will be moved. Defaults to $APS_DIR/out.
APS_ERROR	The directory where all the files will be moved, when there is an error during the processing. Defaults to $APS_DIR/err.
APS_DATA	The directory where all the data files exist. It is intended that this directory is static. Site-specific data (esp. related to configuration) should be located in APS_ETC. Dynamic data will be located in APS_VAR_DATA. Defaults to $APS_DIR/data.
APS_VAR	The directory is for any dynamic files. Defaults to $APS_DIR/var.
APS_VAR_DATA	The directory is for any dynamic data files. Defaults to $APS_DIR/var/data.
APS_LOG	The directory used for logging. Defaults to $APS_DIR/var/log.
APS_LOCK	The directory where all locks are created. Defaults to $APS_DIR/var/lock.

Configuration

The Ruby driver may be configured using an XML configuration file. The Ruby variables reside in a <ruby> element. From here, the user can set the Ruby variables to non-default values for non-standard installations.

The following listing shows all the possible XML configuration values and the defaults or matching environment variable in the case of the directories.

```
<ruby>
  <directories>

      <!-- system locations -->
      <top>$APS_DIR</top>
      <bin>$APS_BIN</bin>
      <lib>$APS_LIB</lib>
      <data>$APS_DATA</data>

      <!-- configuration -->
      <etc>$APS_ETC</etc>

      <!-- active processing locations -->
      <work>$APS_WORK</work>
      <out>$APS_OUT</out>
      <error>$APS_ERROR</error>
      <import>$APS_IMPORT</import>
      <var>$APS_VAR</var>
      <lock>$APS_LOCK</lock>
      <log>$APS_LOG</log>
      <vardata>$APS_VAR_DATA</vardata>

      <!-- input/output data locations -->
      <rs>$APS_DATA_BASE</rs>
      <rs_lvl1>$APS_L1_DATA_BASE</rs_lvl1>
      <rs_lvl2>$APS_L2_DATA_BASE</rs_lvl2>
```

```
        <rs_lvl3>$APS_L3_DATA_BASE</rs_lvl3>
        <rs_lvl4>$APS_L4_DATA_BASE</rs_lvl4>
        <rs_lvl5>$APS_L5_DATA_BASE</rs_lvl5>
        <browse>$APS_IMAG_BASE</browse>
        <browse_lvl1>$APS_L1_IMAG_BASE</browse_lvl1>
        <browse_lvl2>$APS_L2_IMAG_BASE</browse_lvl2>
        <browse_lvl3>$APS_L3_IMAG_BASE</browse_lvl3>
        <browse_lvl4>$APS_L4_IMAG_BASE</browse_lvl4>
        <browse_lvl5>$APS_L5_IMAG_BASE</browse_lvl5>
    </directories>

    <!-- level of logging verbosity -->
    <verbose>1</verbose>

    <!-- number of processing threads (default=1) -->
    <threads>1</threads>

    <!-- realtime processing -->
    <realtime>1</realtime>

    <!-- APS SQL database (default=sqlite:$APS_VAR_DATA/rsdb.db) -->
    <rsdb>pg:host:dbname</rsdb>

    <!-- database scanning options -->
    <dbscan>

        <!-- database scanning switch -->
        <active>1</active>

        <!-- data processing levels -->
        <levels>2,3,4</levels>

        <!-- data sensors (sensor_id or sensor_name)-->
        <sensors>goci,400,modisa,700,800</sensors>

        <!-- data file types -->
        <ftypes>40,55,90,91,100,123</ftypes>

        <!-- process VOCCO data -->
        <vocco>0</vocco>

        <!-- process MODIS L1A to L1B data -->
        <modisl1a>0</modisl1a>

        <!-- check coverage (apsAreas) -->
        <coverage>1</coverage>
    </dbscan>

    <!-- XML configuration -->
    <xml>

        <!-- global.xml (default: $APS_ETC/global.xml)-->
        <global></global>
```

```
        <!-- regions.xml (default: $APS_ETC/regions.xml)-->
        <regions></regions>

        <!-- level-2 processing XML (default: $APS_ETC/l2_gen.xml)-->
        <level2></level2>

        <!-- level-3 processing XML (default: $APS_ETC/l3_gen.xml)-->
        <level3></level3>

        <!-- level-4 processing XML (default: $APS_ETC/l4_gen.xml)-->
        <level4></level4>

        <!-- level-2 image processing XML (default: $APS_ETC/l2_browse.xml)-->
        <image2></image2>

        <!-- level-3 image processing XML (default: $APS_ETC/l3_browse.xml)-->
        <image3></image3>

        <!-- level-4 image processing XML (default: $APS_ETC/l4_browse.xml)-->
        <image4></image4>
    </xml>

    <!-- data transfer options (download/upload)  -->
    <transfer>

        <!-- number of data transfer threads (default=0 -- meaning OFF!) -->
        <threads>1</threads>

        <!-- transfer SQL database (default=sqlite:$APS_VAR_DATA/transfer.db)
 -->
        <db>pg:host:dbname</db>

        <!-- transfer configuration (default=$APS_ETC/downloads.xml) -->
        <xml>$APS_ETC/downloads.xml</xml>
    </transfer>

    <!-- import data? 0=no, 1=yes (default=0) -->
    <import>1</import>

    <!-- vicarious calibration options  -->
    <vical>

        <!-- perform vicarious calibration? 0=no, 1=yes (default=0) -->
        <active>0</active>

        <!-- vicarious calibration configuration (default=$APS_ETC/vc-
aeronet.plist) -->
        <config>$APS_ETC/vc-aeronet.plist</config>
    </vical>

    <!-- polling time in seconds (default=60) -->
    <polling>60</polling>

    <!-- watch dog thread in seconds (default=300*threads/polling) -->
```

```
    <watchdog>5</watchdog>

    <!-- email errors to this address -->
    <email></email>

    <!-- log file -->
    <logfile>$APS_LOG/aps.$(hostname).log</logfile>

    <!-- PID file -->
    <pidfile>$APS_LOCK/aps.$(hostname).pid</pidfile>

    <!-- processing file -->
    <process>$APS_LOCK/aps.$(hostname).process</process>
  </ruby>
```

The following is an example Ruby XML file. In this case, the remote sensing data resides on the NFS share `/rs` and the browse imagery on NFS share `/browse`. The host system is a 4 dual-core linux box so 8 threads will be used by the ruby controller. The system also has a large RAM disk mounted on `/tmp` which be used as the work directory to increase processing speed (avoiding disk IO on temporary files). This is similar to setting APS_WORK to `/tmp/work_APS`.

```
<!-- create some entities to make edits easier -->
<!DOCTYPE ruby [
<!ENTITY % rsdir "/rs">
<!ENTITY % browsedir "/browse">
]>
<ruby>
    <directories>
        <!--  main processing APS directory structures -->
        <work>/tmp/work_APS</work>
        <!--  output data locations -->
        <rs>&rsdir;</rs>
        <browse>&browsedir;</browse>
    </directories>
    <!-- processing parameters -->
    <threads>8</threads>
<ruby>
```

EXAMPLES

Example 1. Output APS Configuration

This example shows the result of using the `verbose` option with `config`. This is very handy to verify that the configuration is set up correctly. First, we will show the contents of the configuration file:

```
$ cat /home/aps/aps_v6.4.4/etc/aps.localhost.domain.config.xml
<!-- create some entities to make edits easier -->
<DOCTYPE ruby [
<ENTITY % rsdir "/home/aps/rs">
<ENTITY % browsedir "/home/aps/browse">
]>
```

```
<ruby>
    <directories>
        <!-- output data locations -->
        <rs>&rsdir;</rs>
        <browse>&browsedir;</browse>
    </directories>
    <!-- processing parameters -->
    <threads>4</threads>
<ruby>
```

Using the configuration file listed above, the `config verbose` command will produce:

```
$ ~/aps_v6.4.4/bin/aps.rb config verbose
APS to use this config file: /home/aps/aps_v6.4.4/etc/
aps.localhost.domain.config.xml
Parameters:
config file      = /home/aps/aps_v6.4.4/etc/aps.localhost.domain.config.xml
verbose          = 1
apsMaxNProcs     = 4
apsPollingTime   = 60
apsImport        = 1
apsRSDataBase    = /home/aps/aps_v6.4.4/var/data/rsdb
apsWatchdogFreq  = 20
File transfer parameters:
transferThreads  = 20
transferDB       = pg:host:dbname
transferXML      = /home/aps/aps_v6.4.4/etc/downloads.xml
Environmental Variables:
APS_DIR          = /home/aps/aps_v6.4.4
APS_ETC          = /home/aps/aps_v6.4.4/etc
APS_BIN          = /home/aps/aps_v6.4.4/i686-centos5.5/bin
APS_LIB          = /home/aps/aps_v6.4.4/i686-centos5.5/lib
APS_IMPORT       = /home/aps/aps_v6.4.4/import
APS_WORK         = /home/aps/aps_v6.4.4/work
APS_OUT          = /home/aps/aps_v6.4.4/out
APS_ERROR        = /home/aps/aps_v6.4.4/err
APS_DATA         = /home/aps/aps_v6.4.4/data
APS_VAR          = /home/aps/aps_v6.4.4/var
APS_VAR_DATA     = /home/aps/aps_v6.4.4/var/data
APS_LOCK         = /home/aps/aps_v6.4.4/var/lock
APS_LOG          = /home/aps/aps_v6.4.4/var/log
File Parameters:
apsLogFile       = /home/aps/aps_v6.4.4/var/log/aps.localhost.domain.log
apsPIDFile       = /home/aps/aps_v6.4.4/var/lock/aps.localhost.domain.pid
apsProcessFile   = /home/aps/aps_v6.4.4/var/lock/aps.localhost.domain.process
apsPreProcess    = /home/aps/aps_v6.4.4/i686-centos5.5/bin/apsPreProcess.sh
apsPostProcess   = /home/aps/aps_v6.4.4/i686-centos5.5/bin/apsPostProcess.sh
Data Base Locations:
APS_DATA_BASE    = /home/aps/rs
APS_L1_DATA_BASE = /home/aps/rs/lvl1
APS_L2_DATA_BASE = /home/aps/rs/lvl2
APS_L3_DATA_BASE = /home/aps/rs/lvl3
APS_L4_DATA_BASE = /home/aps/rs/lvl4
```

```
APS_L5_DATA_BASE = /home/aps/rs/lvl5
APS_IMAG_BASE    = /home/aps/browse
APS_L1_IMAG_BASE = /home/aps/browse/lvl1
APS_L2_IMAG_BASE = /home/aps/browse/lvl2
APS_L3_IMAG_BASE = /home/aps/browse/lvl3
APS_L4_IMAG_BASE = /home/aps/browse/lvl4
APS_L5_IMAG_BASE = /home/aps/browse/lvl5
```

Name

apsFileTransferDB.rb — manages aps SQL file transfer data base

Synopsis

```
apsFileTransferDB.rb
```

Description

This Ruby library **apsFileTransferDB.rb** is a class that communicates with the PostgreSQL data base *transfer*. This data base is used to upload processed data from the aps and download data to the aps. This Ruby script does not actually perform the transfer, but is used to manipulate the *transfer* data base.

Options

-U, --upload <local_file> <URI>
: Register a local file to be uploaded to a URI.

-D, --download <URI> <local_file>
: This will register a URI to be downloaded to a local file.

-A, --process
: This option will cause the downloaded file to be placed into the aps for processing. Note that the file should not be downloaded directly into the `in` directory of the aps. Please use the `ftp` directory.

-a, --addressee
: This option will issue an e-mail to all addressees given once the a file has been uploaded. This will give the recpient a notification of successful upload.

-p, --postproc <command>
: This option selects a post processing option to be performed on the file once downloaded or uploaded. It may be set to **gzip**, **gunzip**, **compress**, **uncompress**, or **tar.gz**. For example, a file may be compressed after downloaded.

Note

For an uploaded file the user may not have permissions or capability to perform the post processing command.

-p, --preproc <command>
: This option selects a pre processing option to be performed on the file once downloaded or uploaded. It may be set to **gzip**, **gunzip**, **compress**, or **uncompress**. For example, a file may be compressed before uploading.

Note

For a downloaded file the user may not have permissions or capability to perform the pre processing command.

Note

If uploading a local file, the pre processing command acts on the local file -- it does not make a temporary copy.

-R, --report <file>
: This option will generate a report with the given`file` filename. The report will be a PDF file which breaks down the entries in the transfer database by week and data type. It provides the number and total size of each week.

-v, --verbose   Make script verbose.

Examples

```
FileTransferDB.rb --upload
 terra.2010229.0817.D.L3_Mosaic.modis.GOM.v08.1000m.Kd_488_lee.png \
                  mailto:zero@dev.null
```

will register the given file to be e-mailed to zero@dev.null.

These commands will register the same file to be transferred to the NRL anonymous ftp server. The second command **apsDownloadManager.rb** actually does the transfer.

```
FileTransferDB.rb --upload
 terra.2010229.0817.D.L3_Mosaic.modis.GOM.v08.1000m.Kd_488_lee.png \
    ftp://ftp.nrlssc.navy.mil/pub/incoming/aps/
terra.2010229.0817.D.L3_Mosaic.modis.GOM.v08.1000m.Kd_488_lee.png
apsDownloadManager.rb
```

Note

A FTP URI should contain the full path name (including filename). Also, in general, the input file should be an absolute path. Otherwise, **apsDownloadManager.rb** will need to be run in the directory where the file is located.

```
FileTransferDB.rb --download http://www7333.nrlssc.navy.mil/docs/aps_v6.4.4/
APS-v6.4.4-Users-Guide.pdf $HOME/aps_users_guide.pdf
apsDownloadManager.rb
```

will add an entry so that the APS Users Guide will be downloaded to the $HOME account and renamed aps_users_guide.pdf.

Name

daylight — determine if sun is up for given time and place

Synopsis

`daylight` *`year day hour min lat lon`*

Description

The program **daylight** is intended to be used by shell scripts when two courses of action must be made: one for night time and the other for daytime. The program will determine the elevation of the sun based on a location and time of day. If the elevation is greater than 20 degrees, the program exits with a status of 1 (day). Otherwise, the exit status is 0 (night).

The time must be given as a four digit year, day of the year, hour and minute of the day in UTC. The location is given by the latitude and longitude of the point of interest. These must given in decimal degrees ranging from (-90.0 to 90.0) and (-180.0 to 180.0) respectively.

Options

-e <elevation>	Used to change the default value (20.0) for the elevation.
-v	Use verbose mode.
--help	Print out a small help page.
--version	Print out version of software and quit.

Name

gregor — converts from day-of-year to month/day and vice-versa

Synopsis

`gregor` *`year`* *`yday`*

`gregor` *`year`* *`month`* *`mday`*

Description

The program **gregor** will convert a day of the year into a month and day or vice versa. The year must be a four-digit year.

Options

-d <n>	Add *n* number of days to input date
-r	Compute the "Reynolds day" for given date. The "Reynolds day" is the Wednesday of each week and corresponds to file naming scheme used by Reynolds for his optimum interpolation sst values.
--help	Print out a small help page.
--version	Print out version of software and quit.

Name

hdf — general manipulation functions on an HDF file

Synopsis

`hdf` *`file.hdf`* [*`cat`* | *`copy`* | *`export`* | *`fattr`* | *`history`* | *`import`* | *`list`* | *`note`* | *`sattr`*] [*`options`*] [*`parameters`*]

Description

The program **hdf** is used to manipulate HDF files (currently it only supports the SDS interface). With this program you can copy an SDS (with or without attributes) from one HDF file to another, dump an SDS to the screen, list all SDSs and file attributes in a format similar to NetCDF's Common Data Language (CDL).

The `file.hdf` parameter is the input file to perform the given action on. There will be a different series of parameters for each action taken. The actions that can be performed on an HDF file are: cat, list, copy.

Actions

cat

The *cat* command is used to dump data from the given data set to stdout. The data will be printed according to its type (char's as string, int's as integers, and float's as floating point numbers). There will be 16 columns across the page for 8- and 16-bit signed/unsigned integers; and 8 columns across the page for 32-bit signed/unsigned interger or 32- and 64-bit floating point numbers. Characters will be printed to 72 columns.

By default, the attributes of the data set will be printed. To prevent this, the `-q` can be used. It will quiet down the output.

copy

The *copy* command is used to copy an SDS from one file to another.

To copy all file attributes from one HDF file to another file, use the comand:

```
hdf from.hdf copy -attr to.hdf
```

To copy only file attributes using a regular expression (for example, only the time attributes) from one HDF file to another file, use the comand:

```
hdf from.hdf copy -attr -re 'time*' to.hdf
```

To copy all attributes from an SDS in one file to an SDS in another file, use the command:

```
hdf from.hdf copy -attr to.hdf from.sds to.sds
```

To copy only specified attributes from one SDS in one file to an SDS in another file, use the command:

```
hdf from.hdf copy -attr to.hdf from.sds to.sds attr1 attr2 ...
```

To copy the attributes from an SDS in one file to an SDS in another file renaming the attributes in the process, use the command:

```
hdf from.hdf copy -attr -rename to.hdf from.sds to.sds attr1 newattr1 attr2
 newattr2 ...
```

To copy an SDS with its associated attributes to another file, use the command:

```
hdf from.hdf copy to.hdf from.sds to.sds sds1 sds2 ...
```

To copy the SDS without the associated attributes add the -noattrs option:

```
hdf from.hdf copy -noattrs to.hdf from.sds to.sds sds1 sds2 ...
```

To rename the SDS's during the copy:

```
hdf from.hdf copy -rename to.hdf from.sds to.sds sds1 new1 sds2 new2 ...
```

export

The *export* command is used to export an object that was imported into the HDF file. Usually the imported file is an image or Encapsulated Post Script file.

fattr

The *fattr* command is used to add, change, or view a file attribute. The command has a single option -nt used to define the number type of the attribute. The default number type is “APS_NT_CHAR8” normally used to append strings attributes.

To add an attribute called `browseList` with a value of “chl_oc2,chl_oc4”, the user would run the command:

```
hdf in.hdf fattr browseList chl_oc2,chl_oc4
```

To view an attribute called “createUser” the user would run the command:

```
hdf in.hdf fattr createUser
```

history

The *history* command is used to get an ASCII dump of the contents of the history of the APS file.

import

The *import* command is used to import a file into an HDF file. This allows the user to embed an image or Encapsulated Post Script file (for example a histogram plot) into the HDF file. The file **export** is then used to extract.

list

The *list* command is used to get an ASCII dump of the contents of the HDF file in a format similiar to the netCDF Common Data Language format. The user can also request the output to be in HTML format.

```
hdf from.hdf list [ type ]
```

The parameter *type* can be either `-ascii` or `-html`. If type is not supplied or understood by **hdf**, then it defaults to `ascii`

sattr

The *sattr* command is used to add, change or view an SDS attribute. The command has a single option -nt used to define the number type of the attribute. The default number type is `APS_NT_CHAR8` normally used to append strings attributes.

To add an attribute called `comment` with a value of “very nice image”, the user would run the command:

```
hdf in.hdf sattr chl_oc4 comment "very nice image"
```

To view an attribute called `comment`, the user would run the command:

```
hdf in.hdf sattr chl_oc4 comment
```

Options

--help	Print out a small help page.
--version	Print out version of software and quit.

Name

imgBathy — create a bathymetry product

Synopsis

imgBathy [*options*] <*source*> [<*dest*>]

Description

This program is used to create a bathymetry product for the given input file. The resulting bathymetry will be appended to <source> written as a 2-D float32 array and will be named “bathymetry”. If the user appends a second file name (<dest>) the “bathymetry” data set will be written to that file using the first one for navigation only.

By default, the input bathymetry file is $APS_DATA/ETOPO2v2.DOS file.

Options

-B isp=<isp>, iep=<iep>, isl=<isl>, iel=<iel>, irp=<irp>, irl=<irl>

Defines a subsection of the input image.

isp the starting sample number

iep the ending sample number

isl the starting line number

iel the ending line number

irp the replication factor along the samples dimension(not implemented)

irl the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image.

-B nlat=<nlat>, slat=<slat>, wlon=<wlon>, elon=<elon>, irp=<irp>, irl=<irl>

Defines a geographical subsection of input image.

nlat the latitude of most North Western point

slat the latitude of most South Eastern point

wlon the longitude of most North Western point

elon the longitude of most South Eastern point

irp the replication factor along the samples dimension(not implemented)

irl the replication factor along the lines dimension(not implemented)

	The irp/irl indicates the number of samples/lines to skip or repeat (see previous -B description).
-f <bathymetry_file>	This option is used to specify the input bathymetry file.
-n <name>	This option is used to specify another name for the output data set. The default is bathymetry
-o name=<name>,format=<format>	Define output file. name — name of the output file format — format of the output file
-v	Verbose output.
--help	Print out a small help guide.
--version	Print out version of software and quit.

Environment Variables

APS_DATA The directory where all the data files exist. Defaults to `$APS_DIR/data`.

Files

`BATHY.DAT` The bathymetry file. It is the ETOP5 2-minute gridded product created by NGDC. Each value is in whole meters. The file used by APS originated with SeaDAS.

Examples

In this example, the bathymetry is appended to the given level-3 data file.

Example 2. Adding a bathymetry product to a file

```
$ imgBathy S2000065175121.L3_HNAV_MSB
```

Now, suppose that instead we want to put the bathymetry in a second file called GOM_BATHY.hdf and we want to call the array depth instead of bathymetry.

Example 3. Adding a depth product to a file

```
$ imgBathy -n depth S2000065175121.N3_HNAV_GOM GOM_BATHY.hdf
```

Name

imgBrowse — create a quick-look image

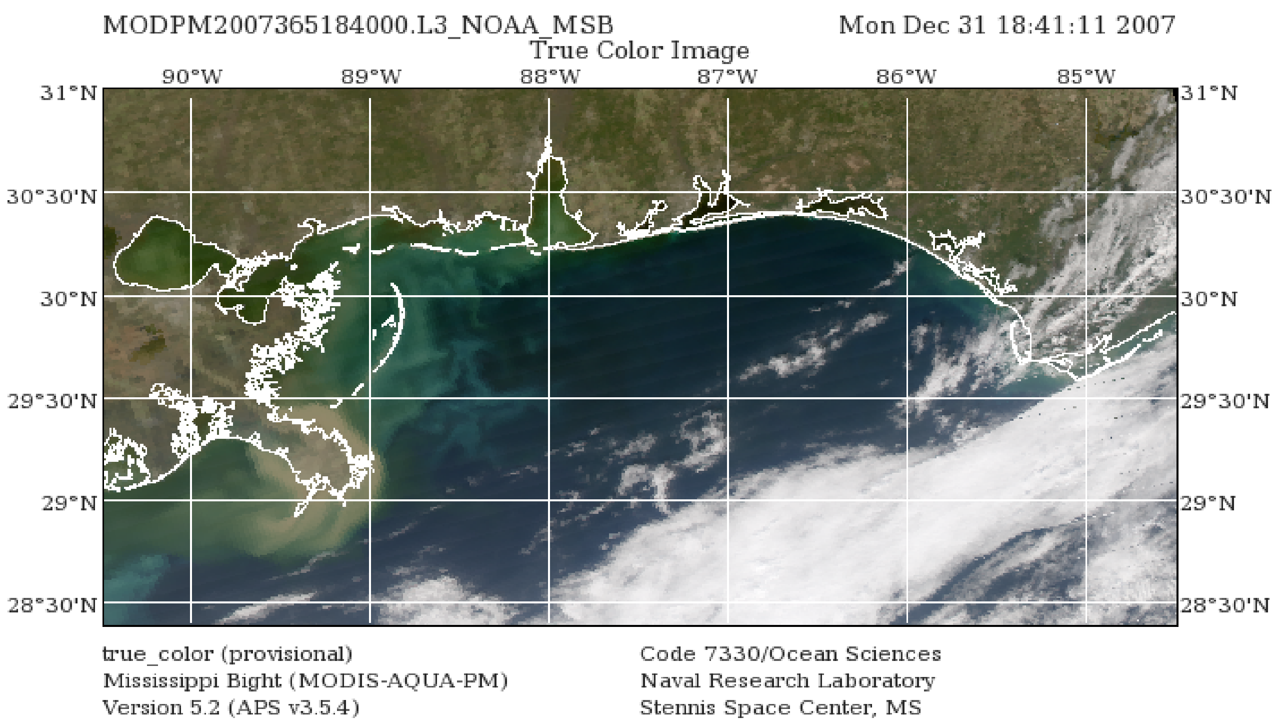
Synopsis

`imgBrowse` [*options*] *<ifile>* *<product>[:<product>:<product>]* *<ofile>*

Description

This program creates a reduced image of the given product(s) in a standard graphic format. The exact formats available depend on the software build process as it uses external libraries. Use the *--help* option to see which formats are known.

Figure 3. Output Image



See the EXAMPLES section below for many different calling ideas. Also see the CAVEATS section below for current problems.

Options

-1	Builds the image at 1-to-1 "full" resolution; do not try to resize the image to a reasonable size. A built-in limit of 4096x4096 should not be exceeded. This option should not be used with -s or -2 options (it will override). If irp/irl are used with -B option, this option should be set.
-2	Builds the image at half resolution. The input image should be less than 4k per side. This option should not be used with -s or -1 options (it will override).
-a <fontfactor>	Used to increase the size of the fonts by multipling the font size calculations (for FreeType rendered fonts or Enscapsulated PostScript

output only). For example, a value of 1.2 will make all fonts 20% larger than normal. The factor is applied equally to all text. The *fontfactor* value must be greater than zero. THIS SHOULD BE THE FIRST OPTION SPECIFIED.

-A <fontpath>	Defines the font path for True Type fonts or the font face for PostScript (e.g., *Helvetica* (default), *Times*, etc.) If the file $APS_DATA/imgBrowse.font exists and True Type fonts are being used, then this file is used.
-b <filename>	Reads in the given blotch file and overlays each region over the image.
-B isp=<isp>, iep=<iep>, isl=<isl>, iel=<iel>, irp=<irp>, irl=<irl>	Defines a subsection of the input image. isp the starting sample number iep the ending sample number isl the starting line number iel the ending line number irp the replication factor along the samples dimension(not implemented) irl the replication factor along the lines dimension(not implemented) The irp/irl indicates the number of samples/lines to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image. Note that by default imgBrowse will try to resize the image to fit a computer screen, so that if the user sets irp/irl the results might not be as expected. The -1 option will force imgBrowse to use the defined sizes.
-B nlat=<nlat>, slat=<slat>, wlon=<wlon>, elon=<elon>, irp=<irp>, irl=<irl>	Defines a geographical subsection of input image. nlat the latitude of most North Western point slat the latitude of most South Eastern point wlon the longitude of most North Western point elon the longitude of most South Eastern point irp the replication factor along the samples dimension(not implemented) irl the replication factor along the lines dimension(not implemented) The irp/irl indicates the number of samples/lines to skip or repeat (see previous -B description).
-c <options>	Sets options that control the colorbar, where: num_labels the number of labels (> 1)

label_format	C-style printf format
ct_num	the colortable to use
lut	file containing a colortable
location	where to place the color ('bottom' for bottom, 'left' for left of image)
reverse	reverse the colortable
brighten	add/subtract bias to/from colortable
bad	use to set the bad data value color (default is white)
invalid	use to set the no data value color (default is black)
background	use to set the background color (default is white)
name	use the name of the color table

The keywords *reverse* and *brighten* must *follow* the color table entry (either *ct_num* or *lut*).

The current available colortables are

0	nrl	NRL Rainbow
1	bios	NASA Bio Sphere
2	ndvi	NASA NDVI
3	diff	Blue to Red
4	neg	Negative Radiance
5	chlor	Goddard Chlorophyll-a
6	vis	Visibility
7	sst	Sea Surface Temperature
8	jhu-sst	Johns Hopkins Univ. APL SST
9	hue2	Hue 2
10	rainbow	Rainbow
11	purple-red	Purple Red Strips
12	vis2	NRL Visibility
13	eos	EOS colortable
14	models	Models colortable
99	gray	Greyscale

The *lut* table refers to a colortable in either the SeaDAS format (`.lut`), the NSIPS format (`.ct`), or XVision format (`.pal`).

Example of option usage: **-c sst,brighten=2**

-C <num>

Sets colortable (see table above).

-C product=<name>, range=<n:m:l>, scale=<n.m>, offset=<n.m>, netcdf=<file>

Sets the contour overlay options, where:

netcdf	NetCDF file containing data to produce contours for. Must contain 1 or 2-dimensional latitude and

	longitude arrays describing the navigation at every point.
hdf	APS file containing data to produce contours for. Must contain APS compatible navigation.
bathy	ETOPO bathymetery file to produce elevation contours for.
product	name of product to produce contours for
timestep	0 for first timestep, 1 for second, ... etc. (most significant dimension of 3 or 4 dimensional product)
depth	0 for first depth, 1 for second, ... etc. (second most significant dimension of 4 dimensional product)
range	3 element, colon separated declaration of contour points. the first element is the number of contour points, and the second and third elements are the minimum and maximum contour points respectively. The contour points will be evenly spaced between the minimum and maximum.
contour_point	4 element, colon separated declaration of an individual contour point. This option may be used separately or in addition to the range option. The first element is a contour point value, the second through fourth elements are the Red, Green, and Blue respectively describing the color for the contour. If this option is used after a range declaration, the contour and color will be appended to the current list of contour points.
invalid	set a value that will be ignored as invalid when scanning the product data.
scale	scale to modify the input data
offset	offset to bias the input data
units	string representing specific units
latitude	set latitude product name
longitude	set longitude product name
lonlat	switch notion of lonlat ordering
interpolate	set to zero to prevent lat/lon interpolation
drawpoints	set to greater than 0 to turn vertice drawing on
verbose	increase verbosity
wlon	the western most longitude for bathy option

elon	the eastern most longitude for bathy option
nlat	the northern most latitude for bathy option
slat	the southern most latitude for bathy option
isp	the starting sample number
iep	the ending sample number
isl	the starting line number
iel	the ending line number
subsamp	subsampling ratio for input data
labels	draw labels (1) or do not (0)
precision	set number of lines to use to draw curve between consecutive contour vertices
frac	set curviness of contour curves. should be between 0 and .5, lower numbers produce sharper curves, larger numbers produce curvier curves.
line_width	set line width of drawn contour lines

The color table used for contour drawing is set in the usual way (i.e. by using the -C option with a number, or by setting the color table options). The color table should be set before defining the contour parameters. Different color tables may be used for each contour by choosing a color table between each contour statement. The default color table may be reset by setting the color table to -999 (i.e. -C -999).
-C range=21:-.5:.5,units=meters,product=Surface_Elevation,netcdf=file

This will read the product "Surface_Elevation" from the file "file" and create contours at 21 levels spaced evenly between -.5 and .5. The latitude and longitude arrays will be read from their default names of Latitude and Longitude respectively.

-d

Turns on debug output (may be very verbose)

-E

Adds text indicating this product is "EXPERIMENTAL". Additionally if there is a file named experimental.png in $APS_DATA, it will use that file as the watermark (-w overrides this behavior).

-f <linear|log|log10>

Selects Function used for scaling (defaults to *linear*).

-g draw=<0|1>color=<R:G:B|name>spacing=<f>, lat_spacing=<lat>, lon_spacing=<lon>, lat_label=<1|2|3>lon_label=<1|2|3>lat_label_color=<R:G:B|name> lon_label_color=<R:G:B|

Sets grid overlay options, where:

draw	turn on (1 - default) or off (0) grid lines
color	color (0.0 - 1.0) used to draw grid lines. May be defined as RGB triplet or named color.

name>lat_label_modulo=<n>, lon_label_modulo=<n>, line_width=<w, sytle=<0|1|2>label_format=<dms|dd>

spacing	lat/lon spacing between grid lines
lat_spacing	spacing between latitudinal grid lines
lon_spacing	spacing between longitudinal lines
lat_label	1 - for labels on left of image 2 - for labels on right of image 3 - for labels on both sides of image
lon_label	1 - for labels on left of image 2 - for labels on right of image 3 - for labels on both sides of image
lat_label_color	set color of latitude labels
lon_label_color	set color of longitude labels
lat_label_modulo	skip n grid lines for latitude labels
lon_label_modulo	skip n grid lines for longitude labels
line_width	set the width of the grid lines (default=1.5)
style	line style: 0-solid, 1-dashed, 2-dotted
label_format	format of labels (dd for decimal degrees) (dms for degrees, minutes, seconds)

For example, **-g color=green,spacing=0.5,style=2** draws green gridlines spaced every 0.5 degrees."

-G <dir>

Name of the input data directory. Defaults to $APS_DATA.

-i

Create the image only. In this case the border, the color table, and annotations, etc. will not be drawn. However, the logo and watermark are still drawn.

-I interlace=<0|1|2>band=<0>, flip=<0|1|2|3>, scale=<f>, offset=<f>, inverse, geofile=<file>

Sets options for the input image, where:

interlace	the interlace mode for the 3-D input data set (0: BSQ, 1: BIL, 2: BIP)
band	selected band to display
flip	flag to flip the input image (1: flip vertically, 2: flip horizontally, 3: flip both vertically and horizontally)
scale_factor	scale the input data
add_offset	add offset to input data
inverse	take inverse of input data
geofile	use given file to find geo-location data

The scale and offset are useful for simple linear transformations of the input data. The inverse option is useful for converting attenuations (1/m) into lengths (m).

-j <file>	Sets the bathymetery option for overlaying a 30m bathymetry contour over image using the given input file. See imgBathy and imgMakeLatLon on how to create this file. The function **apsMakeBathy** in the **apsScripts.sh** file automatically creates the appropriate file.
-J	Used to overlay the met/ozone data used by the file over the image. The MET wind data is overlayed as vectors and the OZONE data is overlayed as contours.
-k	Defines the location (if any) of the classification of the image. May be set to 0 (no classification) or 1 (classification written to top and bottom on image).
-K <options>	This option is used to create a colorbar consisting of breakpoints. That is data between breakpoints will be all set to the same color. Each break point is separated by semicolons (;) and the color is set after the break point with a colon (:). Each component of the color (red,green,blue) are separated by commas (,). For example, **-K 0:0,0,0;20:1,0,0;40:0,1,0 -r 0,40 ifile sst sst.jpg** will create an image of sst such that all pixels less than 0 will be black. Those from 0 to 20 will be red, those from 20 to 40 will be green.
-l	Sets the options used to control the landmask, where: draw — draws (1) or does not (0) the land mask color — defines the color of the landmask file — sets the name of the input landmask file The input landmask file may be the default SEADAS file `$APS_DATA/landmask.dat` or an NSIPS created landmask file.
-L <file>	Sets the logo file
-L file=<file>,width=<w>,height=<h>,x=<x0>,y=<y0>	Sets the options used to control the logo, where: file — sets the name of the input logo file xsize|width|w — sets the width of the displayed logo ysize|height|h — sets the height of the displayed logo x — sets x-location of the displayed logo y — sets y-location of the displayed logo
-m <mapFile:mapName>	Use the following mapFile and mapName for navigation.
-M draw=<0|1>, name=<name>, masks=<"NAME1;NAME2;NAME3">	Controls how mask flags are overlaid on the image. draw — whether to draw masks: 0-no, 1-yes (default is 1). name — name of mask file masks — names of masks to overlay. Multiple mask names must be separated by a semi-colon ";". A color may be assigned for each

mask by following the mask name with a colon ":" and an RGB triplet or color name separated by commas.

Examples:
-M mask="CLDICE;ATMFAIL"

or
-M masks="CLDICE:0.8,0.8,0.8;ATMFAIL:red"

-N <file>	Read in file and place these notes at the bottom on the image. This file is a simple UNIX text file and each line is written verbatum to the bottom on the image.
-o format=<format>, compress=<d>, quality=<85>, level=<3>, transcolor=<R:G:B|name>	Sets options that control the output file, where: Keyword — Description format — output format of file (e.g. jpg) compress — TIFF file compression ('j' for JPEG, 'd' for deflate) quality — JPEG compression quality (1-100, defaults to 85) level — deflate compression level (1-9, defaults to 6) transcolor — color to use for transparency The format is a typical file extension, like jpg or png. An alpha channel will only be written to a format that can handle one. Presently, this option is limited to TIFF and PNG formats.
-O <x>	Sets the offset in the log10 function.
-p <file>	Place points or symbols from the file on the image (see FILES below).
-P	Get political boundaries from $APS_DATA/polbnd.dat in Piskor's format and overlay them on the image.
-Q <file>	Creates a thumbnail file in JPEG format. This thumbnail is a small version of the output image without all the annotation and border.
-r <min,max>	Select range of input data for scaling. Defaults to min/max of each product's validRange parameters. Failing that it uses the min/max of product. Failing that it is set to min/max of all reals.
-s <w,h>	Sets the size of output image. Used to reduce or enlarge the size of the output. Defaults to size of original image. Cannot be used with the -B or -1 options.
-S <slope,intp>	Set the slope and intercept to use for scaling the image
-t file=<file>, color=<R:G:B|name]>, line_width=<w>, font_size=<f>, label=<label>, skip=<n>, delim=<d>,	Set options that provide ship track overlays, where: file — pathname of the ship track file

cols=<lat:lon>, points=<0|1]>, symbol=<n>

delim	set the delimeter between columns in file. Default is tab.
cols	set columns which contain the lat/lon (0-relative)
skip	number of records from top of file to skip
color	set desired color for ship track file
line_width	set the width of the ship track (default=1.0)
font_size	set size of label
label	set ship track label (default=filename)
no_label	not draw label
points	set to 1 to draw track as points rather than line-segments
symbols	set to symbol type for draw points

These options allow the user to apply a ship track to the output image. The default track file is a UNIX tab-delimited text file with latitudes in column 1 and longitudes in column 2. The latitudes and longigtudes are expected to be in decimal degress in the range (-90 to 90) and (-180,180), respectively.

If the input file uses a different delimiter, the user can change it using the **delim** keyword. The first **n** lines might be skipped (if for example there is a header), using the **skip** keyword. Lastly, if the latitudes and longitudes are located in different columns, then the user may set them using **cols**, like **'cols=2,3'**.

The other options allow the user to define the color, line width, and labeling for the ship track.

-T title=<title>, date=<date>, units=<units>, sensor=<sensor>, long_name=<long_name>, str1=<str1>, str2=<str2>, str3=<str3>, text=<text:lat:lon:type:scale:r:g:b:>

Modify the default strings for image.

title	to replace file name (upper left)
date	to replace data (upper right)
long_name	to replace long name (upper middle)
units	to replace units
sensor	to replace sensor
str1	to replace top string on lower right before logo
str2	to replace middle string on lower right before logo
str3	to replace bottom string on lower right before logo
text	to add a text symbol to plot

-U <type>

Used to create a world file that describes the navigation of the output file. A world file is a simple text file that will contain either map coordinates (type=1) or geographical coordinates (type=2). The world file will be named after the input file with the addition of the letter **w** (e.g. if the output file is `world.jpg`, the world file will be `world.jpgw`).

-V nogaps=<file>, met=<file>, pctides=<file>, timestep=<0|1>, color=<R:G:B>, factor=<n.m>, scale=<n.m>, units=<string>, arrow_px_len=<n>, line_width=<w>

Vector overlay option

nogaps	file containing nogaps wind vectors to overlay
met	file containing NCEP MET wind vectors to overlay
pctides	file containing PCTides current vectors to overlay
ncom	file containing NCOM current vectors to overlay
vfile	for the v-component NCOM vector file.
timestep	0 for first nogaps timestep, 1 for second
depth	0 for first depth, 1 for second, ... etc.
factor	define the representation of one arrow length
units	string representing specific units
scale	scale to modify the input vector data
offset	offset to modify the input vector data
subsamp	subsampling ratio for input data
latitude	sets latitude product name
longitude	sets longitude product name
lonlat	switch notion of lonlat ordering
ucomp	sets U-component product name
vcomp	sets V-component product name
thresh	minium windspeed threshold needed to draw vector
color	set desired color (0.0-1.0) for vector file
arrow_px_len	length of standard vector (default is 20.0 pixels)
line_width	the width of the vector lines (default=1.5)
verbose	increase verbosity

For example, **-V factor=0.25,scale=0.5144,units="m/s",pctides='file',color=0:1:0**, renders pctide data from converted from knots to m/s (0.5144), in green, where a vector of 20 pixels represents 0.25 meters.

-v — verbose output (use more than once to increase programs verbosity)

-W <file> — defines the input coastline file. Defaults to `$APS_DATA/world_01.dat`.

-W file=<file>, draw=<0|1>, color=<R:G:B>, tolerance=<t>	Sets up options used to draw coastline, where:

draw	draw (1) or not draw (0) the coastline
color	set the color of the coastline
file	define the input file of the coastline
tolerance	defines the tolerance to reduce coastline resolution

The *tolerance* is used by the Duglas-Peuker algorithm to reduce the resolution of the input coastline file. This is useful when the input file and output image contain a large region of the earth, by reducing the number of points draw by the graphics library. A value of 0.05 is good for a coarse coastline for the entire world. A smaller value will yield better resolution.

The *file* is used to set the input coastline file. It can be set to an NSIPS derived coastline file.

-w <file>	Use file as watermark image. The image will be scaled to fit the product image and overlayed on top of it. This should be an image created with transparency so that when it is overlayed it will give the illusion of a watermark (recommend creating a white image with alpha channel set to 20-30% opacity).
-x	Prevents the drawing of masks on the image, unless dealing with a true color image, then enables drawing of masks on the image. This option will not override masks set with -M, but this option must be used when dealing with true color images in order for the masks set with -M to be drawn.
-X	Prevents the drawing of the coastline on the image.
-Y prod=<prod>, file=<file>mask=<bits>, invalid, color=<R:G:B>, tolerance=<t>	Replace masked areas with an image from the given product. Useful to replace LAND and CLDICE grey mask with a true color image. Or the NDVI over LAND pixels.

prod	name of product to overlay on image
file	name of file containing product if not in source file
mask	defines the bitmask used to determine location of overlay image pixels
invalid	overlay image over product's invalid pixels
ct_num	color table to use for overlay product
lut	color table file to use for overlay product
reverse	reverse the color table
brighten	brighten color table
skip	overlay product reduction to use (if overlay product is not same size as source product)

-Z <file>	Read in *file* to set options

--help	Print out a small help guide.
--version	Print out version of software and quit.

Files

`$APS_DATA/browse/world.dat`	This is the default coastline file. It is part of the Naval Satellite Image Processing System (NSIPS).
`$APS_DATA/common/landmask.dat`	This is the default landmask file. It is part of the SeaWiFS Data Analysis System (SeaDAS).
`$APS_ETC/logos/imgBrowse.logo`	This is the default logo file which contains the default logo to apply to the image. This can be a symbol link to a logo file. See the `-L` option for default formats.
`$APS_DATA/browse/polbnd.dat`	This is the default political boundry file. Only used when `-p` is used.
`$APS_DATA/browse/experimental.png`	This is the default watermark file that is used by the `-E` option to write the words experimental over the image.
`symbols.dat`	This file allows the user to define and apply symbols to the image. The format is white space delimetered UNIX text file in columnar format. The first two columns are the latitude and longitude in decimal degrees with negative being South or West. The next column contains the symbol type: 0 for filled box, 1 for outline of box, 2 for big filled box, 3 for big unfilled box, 5 for a star). The next column is a scale parameter to increase/decrease the size of the symbol. The next three columns describe the color as an RGB triplet given in floating point values (i.e, 0.0 to 1.0). Starting at text column 42, the rest of the line is used as a text string which will be printed next to the symbol. A '#' as the first character of a line indicates a comment and the line will be skipped.
`track.dat`	The track file allows the user to define and apply ship track or other transect over the image. The file format is a white space delimetered UNIX text file in columnar format. The only two columns are the latitude and longitude in decimal degrees with negative being South or West. A '#' as the first character of a line indicates a comment and the line will be skipped.
`blotch.dat`	The blotch file allows the user to define and apply polygon areas over the image. The file format is a UNIX text file which begins with a single line containing the number of blotches (*n*) in the file. This is followed by *n* groups of lines for which the first line contains the name of the region and the second line contains the number of longitude, latitude (*m*) pairs in the polygon. The next *m* lines contain two space delimeted columns containing the longitude and latitude of the points on the polygon. All polygons are assumed to be closed. A '#' as the first character of a line indicates a comment and the line will be skipped.
`vector files`	The vector overlay function can handle files produced by the PC-TIDES, NOGAPS, NCOM, and SeaWiFS/MODIS climatology MET files.
`contour files`	A contour file must contain the following attributes....

Environment Variables

$APS_DATA This environmental variable should point to the APS's data directory. It is used to find static data files like the coastline file.

$APS_ETC This environmental variable should point to the APS's etc directory. It is used to find user configuration data files like the logo.

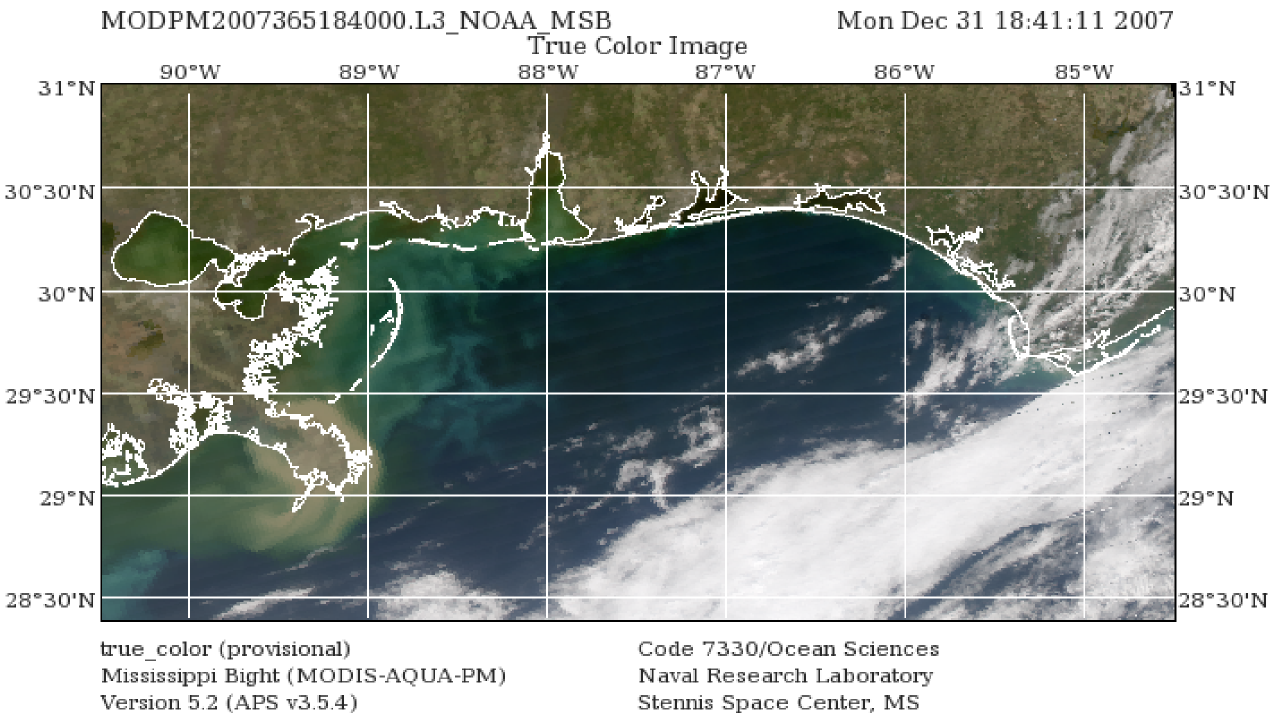
Examples

To create a true color image with coastline overlays and other proper annotation from the file S20000001175612.L3_HNAV.

Example 4. Creating a TIFF true color image

```
$ imgBrowse S20000001175612.L3_HNAV rhos_670:rhos_555:rhos_443
 S20000001175612_true_color.tiff
```

Figure 4. True Color Image

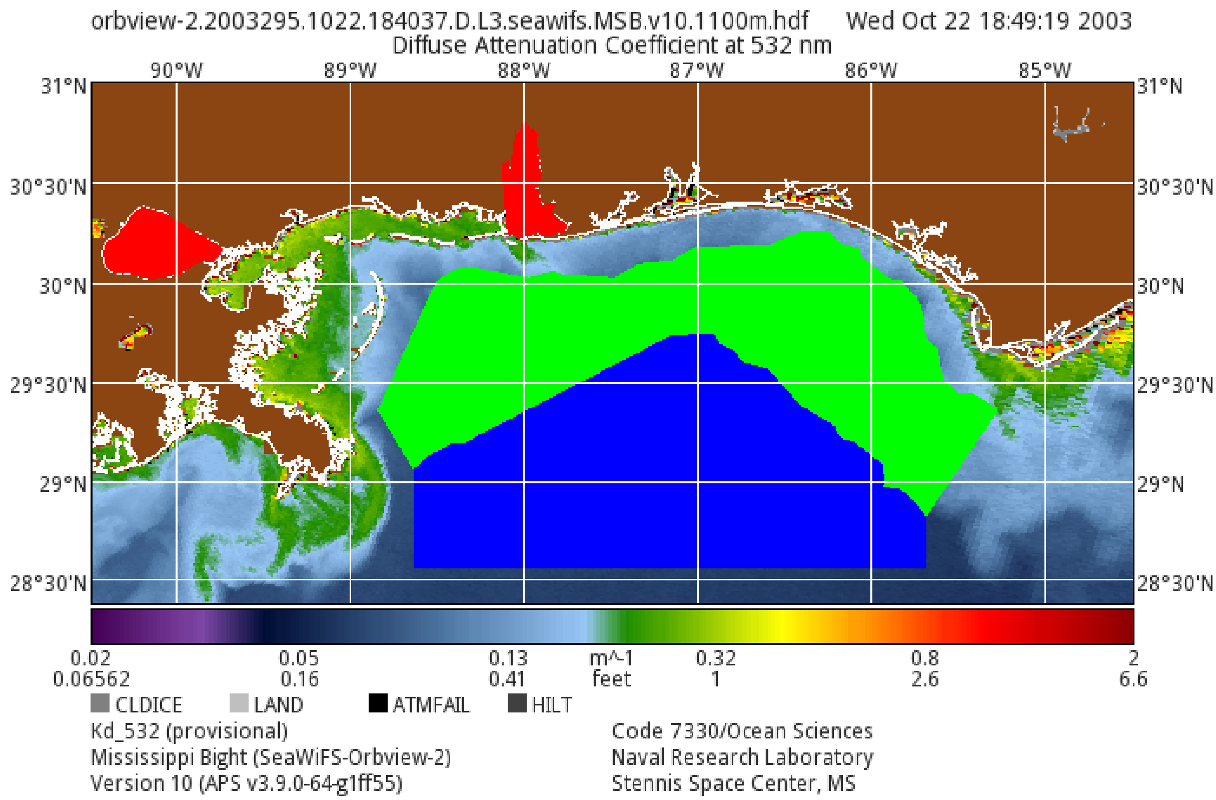


Example 5. Overlaying a region of interest on an image

A region of interest (or blotch) is a general geographical polygon that used by APS for the time series data extraction capability using **imgTSeries**. The `-b` can be used to create an image in which those polygons are overlaid on the image. For example,

```
$ imgBrowse -b MissBight.blotch
 orbview-2.2003295.1022.184037.D.L3.seawifs.MSB.v10.1100m.hdf Kd_532
 MissBight_blotch.png
```

Figure 5. Region-of-interest Overlay on an image

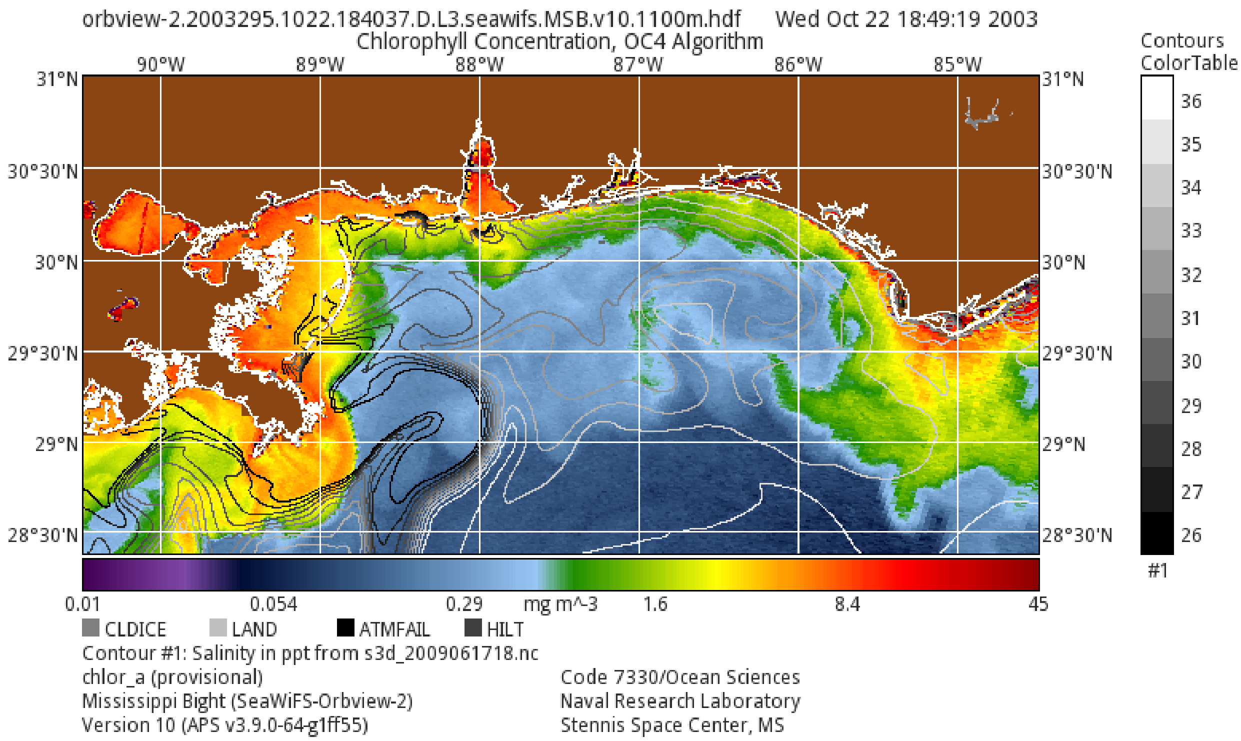


Example 6. Overlay contours on a browse image

The `-C` allows the user to overlay contours over an image. In this example, modeled salinty values from 26 ppt to 36 ppt are overlaid on a chlorophyll-a image.

```
$ imgBrowse -C -999 -C
 line_width=1,units=ppt,product=Salinity,range=11:26:36,netcdf=s3d_2009061718.nc
 orbview-2.2003295.1022.184037.D.L3.seawifs.MSB.v10.1100m.hdf chlor_a a.png
```

Figure 6. Salinty Contours Overlaid on Image

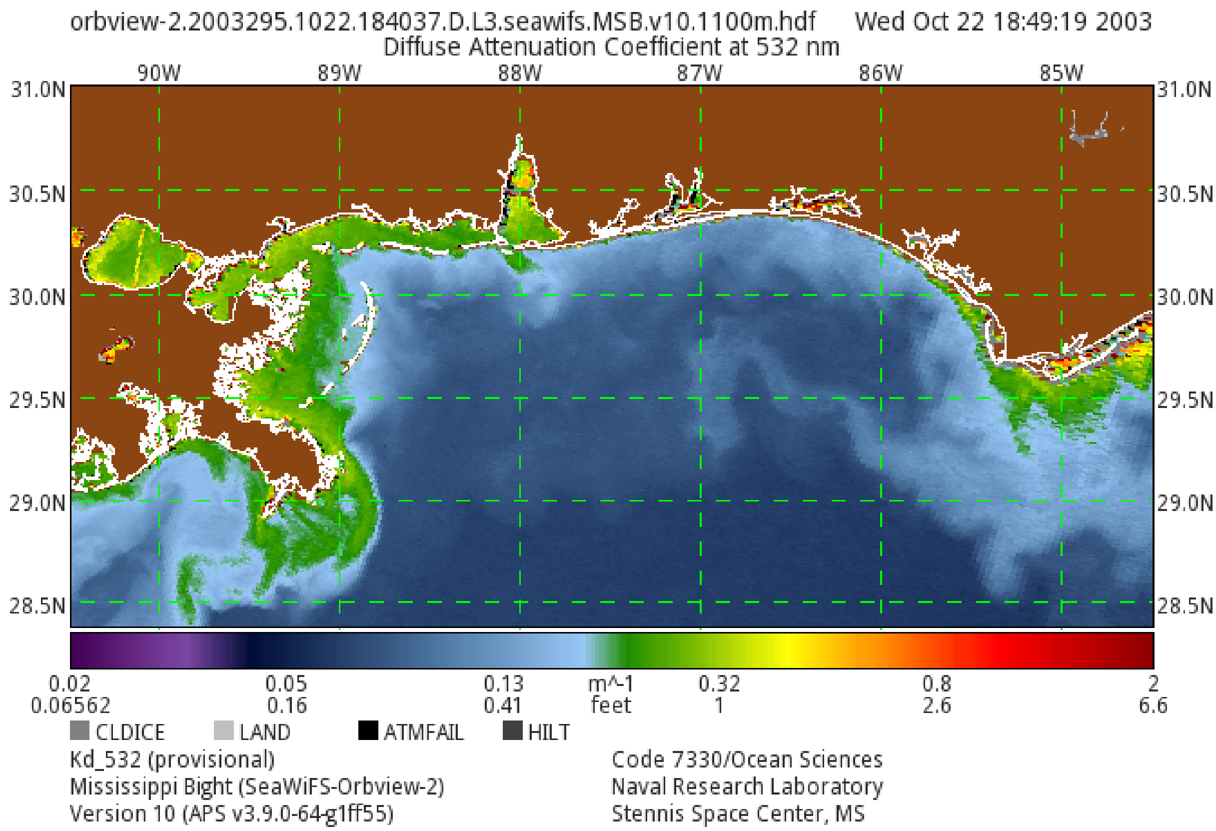


Example 7. Change grid overlays on image

The `-g` provides options to enhance the default grid options. For example,

```
$ imgBrowse -g color=green,style=1,label_format=dd
 orbview-2.2003295.1022.184037.D.L3.seawifs.MSB.v10.1100m.hdf Kd_532 grids.png
```

Figure 7. Output Image

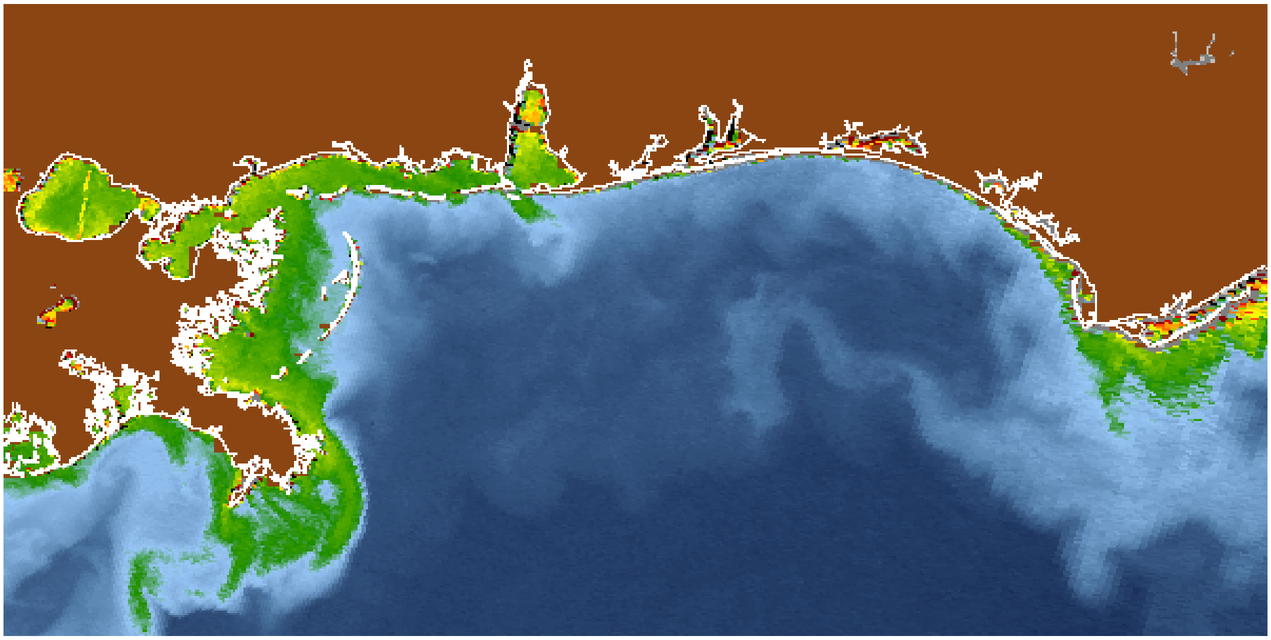


Example 8. Create a browse image with no annotations or borders (only satellite data)

The `-i` restricts the output to the image only and does not provide the border and any text annotations. However, grids, coastlines, masks, etc. are still available. In this example, the grids are turned off using the `-g draw=0` option.

```
$ imgBrowse -i -g draw=0
 orbview-2.2003295.1022.184037.D.L3.seawifs.MSB.v10.1100m.hdf Kd_532
 image_only.png
```

Figure 8. Image Only

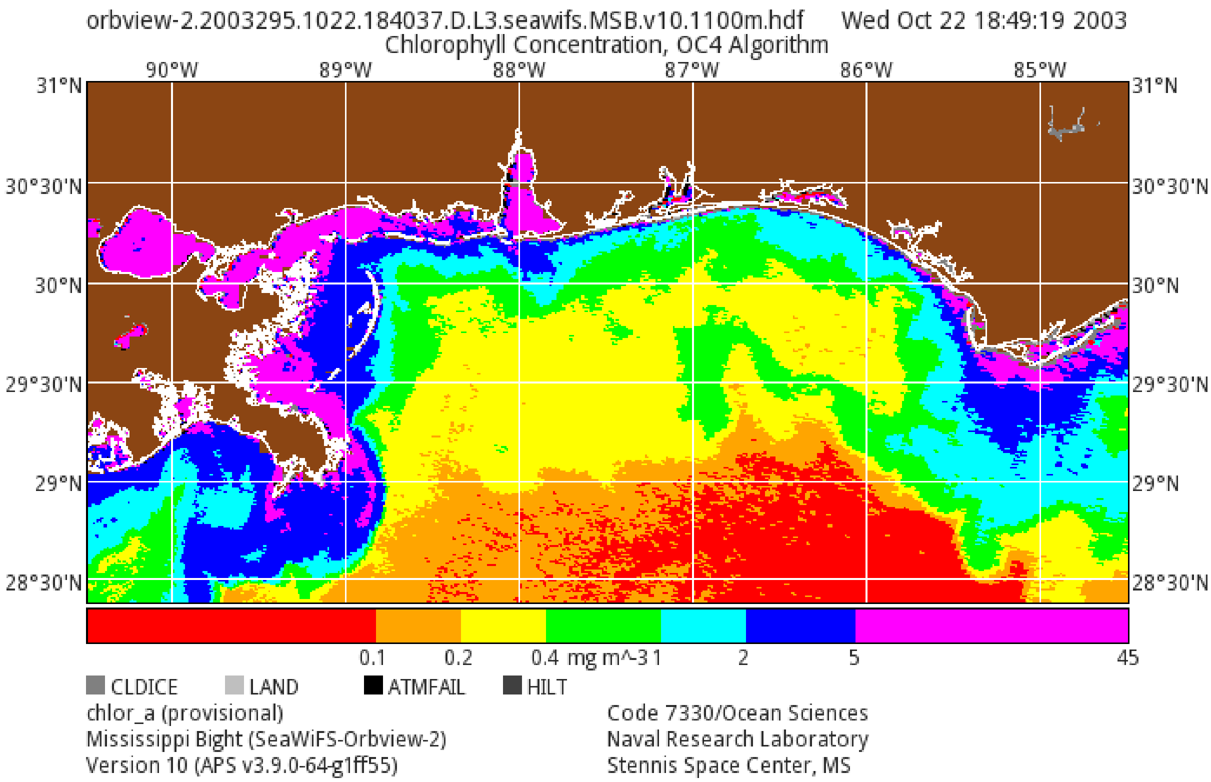


Example 9. Creating browse image using a discrete color table

The `-K` allows the user to set a discrete color table consisting of “breakpoints” that are used to define the colors based on data values. In this example, the chlor_a product is shown with discrete values.

```
$ imgBrowse -
K'0:black;0.1:red;0.2:255,165,0;0.4:yellow;1.0:0,255,0;2.0:0,255,255;5:0,0,255;45:255
 orbview-2.2003295.1022.184037.D.L3.seawifs.MSB.v10.1100m.hdf chlor_a
 discrete_colors.png
```

Figure 9. Discrete Colors



Example 10. Adding annotations to a browse image

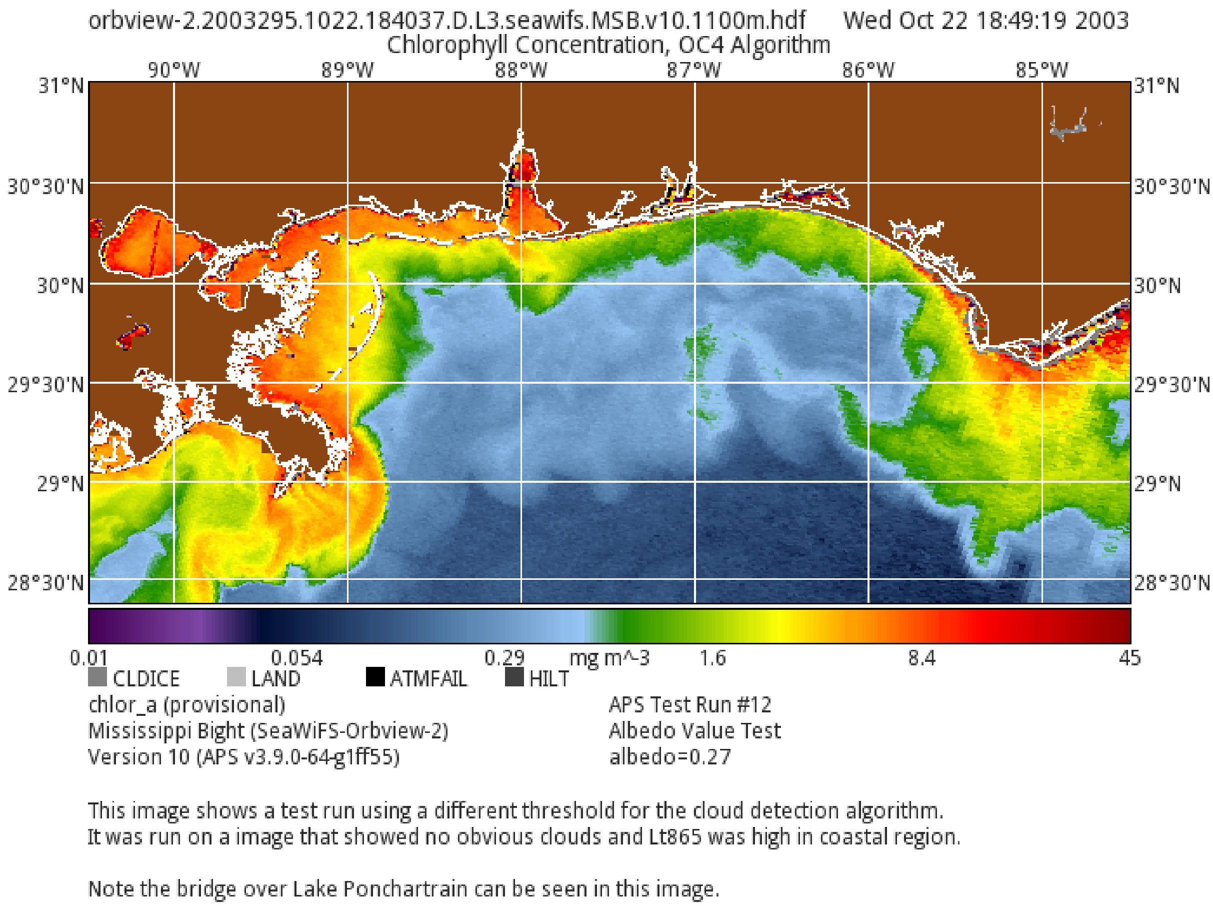
The `-N` and `-T` options allow the user to define text that will be displayed on the image. In this example, a short notes file was created indicating this was an albedo test image. The strings right side provide the values used during the test.

```
$ cat notes.txt

This image shows a test run using a different threshold for the cloud
 detection algorithm.
It was run on a image that showed no obvious clouds and Lt865 was high in
 coastal region.

Note the bridge over Lake Ponchartrain can be seen in this image.
$ imgBrowse -N notes.txt -T str1='APS Test Run
 #12',str2='Albedo Value Test',str3='albedo=0.27'
 orbview-2.2003295.1022.184037.D.L3.seawifs.MSB.v10.1100m.hdf chlor_a
 notes.png
```

Figure 10. Adding notes and annotations to a browse image

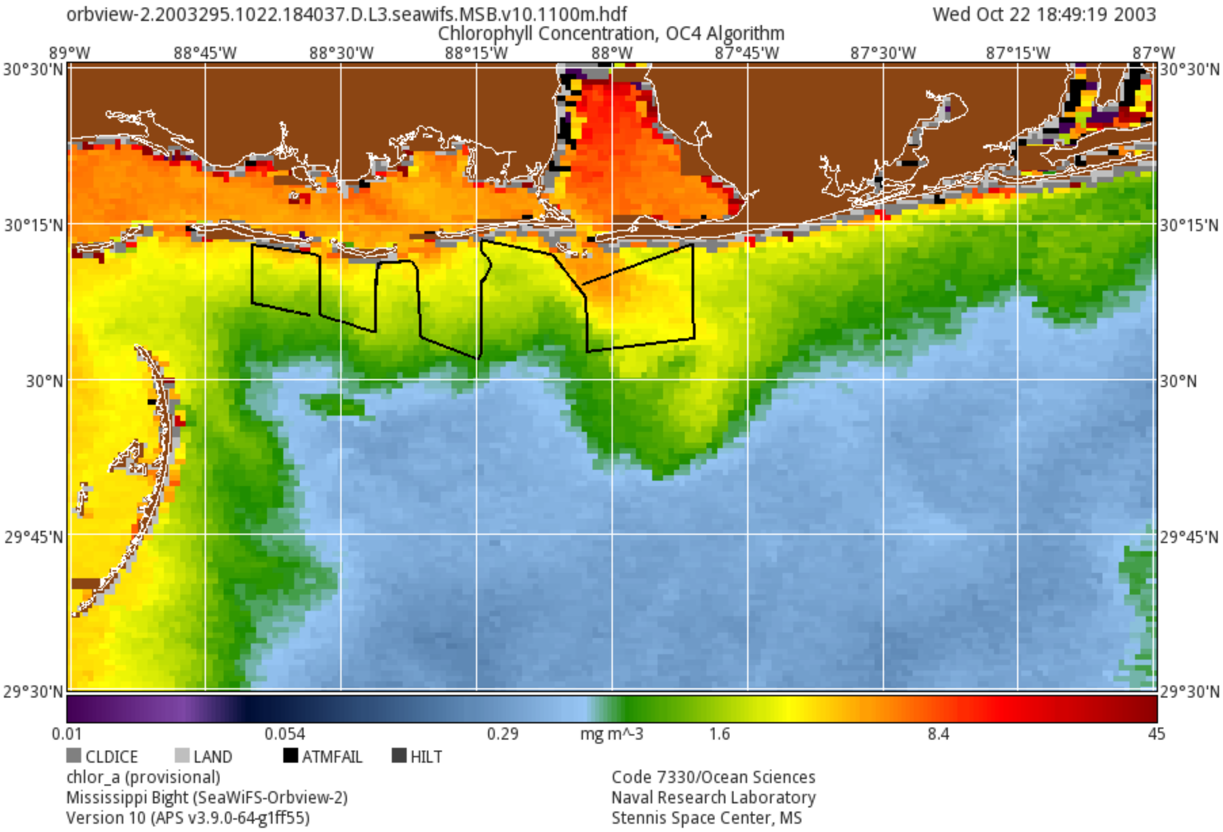


Example 11. Overlay a ship track on a browse image

The `-t` allows the user to overlay (ship) track over an image. In this example, a ship track from a Miss Bight cruise is overlaid on the image in black. To zoom into the area of the original image covered by the ship, the `-B` option is used. The `spacing` option for grids are also set to every 15 minutes (or quarter of a degree).

```
$ imgBrowse -B nlat=30.5,slat=29.5,elon=-87,wlon=-89 -g spacing=0.25
 -t file=~/MSB052202_track.txt,color=black,no_label,line_width=2
 orbview-2.2003295.1022.184037.D.L3.seawifs.MSB.v10.1100m.hdf chlor_a
 track.png
```

Figure 11. Ship Track Overlaid on Image

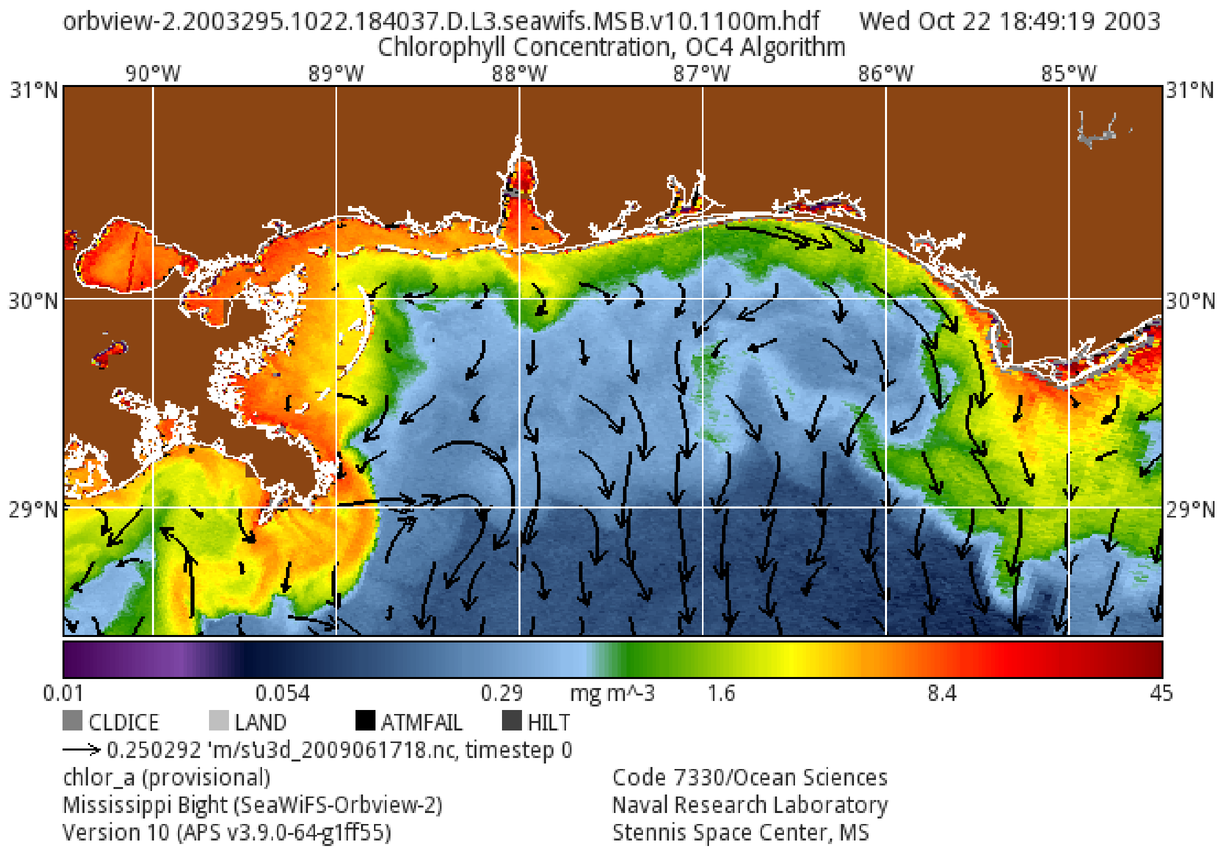


Example 12. Overlay a set of vector over a browe image

The `-v` allows the user to overlay vectors over an image. In this example, “curvy” vectors from a model are overlaid on a chlorophyll-a image.

```
$ imgBrowse -g spacing=1 -V
 color=black,line_width=2,thresh=.05,subsamp=8,factor=.832903,ucomp=U_Velocity,vcomp=V_V
\'m/s\',vfile=v3d_2009061718.nc,curvy=u3d_2009061718.nc
 orbview-2.2003295.1022.184037.D.L3.seawifs.MSB.v10.1100m.hdf chlor_a a.png
```

Figure 12. Vectors Overlaid on Image



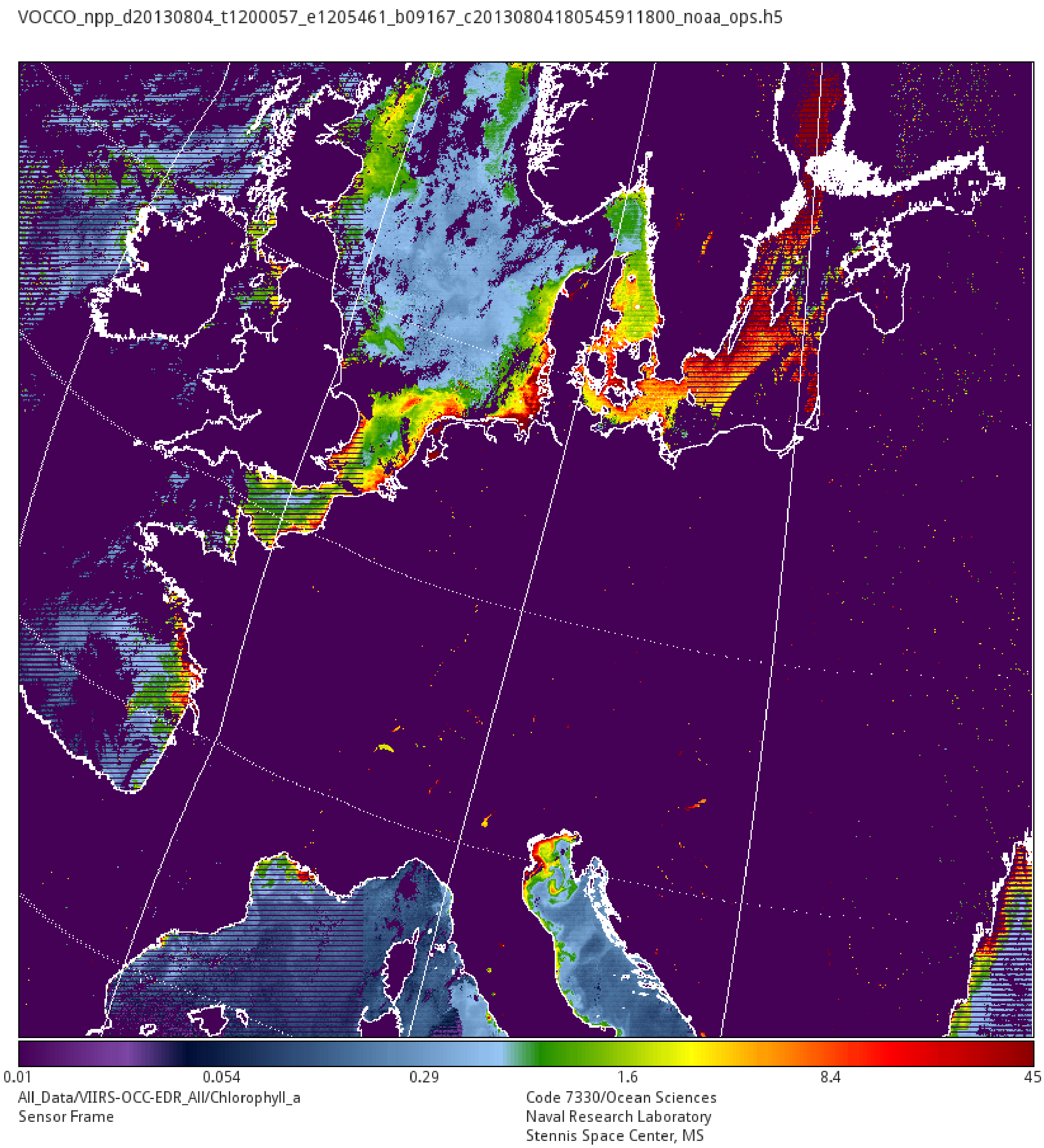
Example 13. VIIRS OCC EDR Chlorophyll-a Image

In this example, we use the -I geofile option to create an image of the Chlorophyll-a product from the VIIRS OCC EDR file.

```
$ bin/imgBrowse -1 -I
 flip=3,geofile=GMODO_npp_d20130804_t1200057_e1205461_b09167_c20130804180546567378_no
 \
  -g draw=2 -l draw=0 -r 0.01,45 -f log10 \

 VOCCO_npp_d20130804_t1200057_e1205461_b09167_c20130804180545911800_noaa_ops.h5
  ll_Data/VIIRS-OCC-EDR_All/Chlorophyll_a chlora.png
-W- colortable cannot be obtained from /rs/lvl2/viirsn/VOCCO/2013/216/
VOCCO_npp_d20130804_t1200057_e1205461_b09167_c20130804180545911800_noaa_ops.h5
 (using default)
VIIRS geo-location samples 3200, lines 3072
-W- unable to open flags l2_flags
-W- unable to open flags l2_flags
```

Figure 13. VIIRS OCC EDR Chlorophyll-a



Example 14. VIIRS OCC EDR Cloud Flags

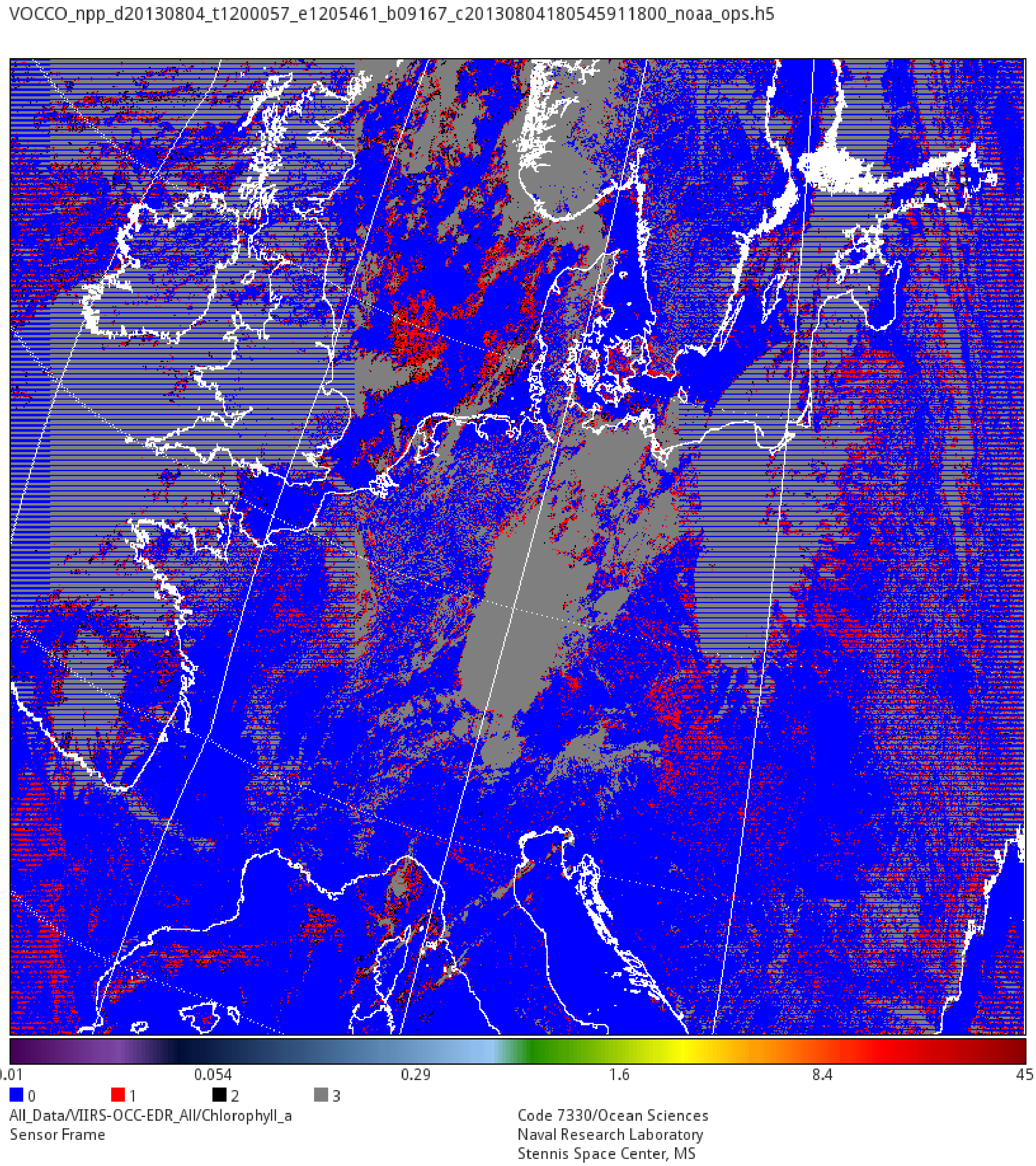
In this example, we use the -M option to create an image of all four cloud bits (Confidently Cloudy, Probably Cloudy, Probably Clear, and Confidently Clear).

```
$ bin/imgBrowse -1 -I
 flip=3,geofile=GMODO_npp_d20130804_t1200057_e1205461_b09167_c20130804180546567378_noaa_
 \
```

```
  -g draw=2 -l draw=0 -r 0.01,45 -f log10 \
  -M name=All_Data/VIIRS-OCC-EDR_All/
QF5_VIIRSOCCEDR,masks="0:blue;1:red;2:black;3:127|127|127",filter=3 \

 VOCCO_npp_d20130804_t1200057_e1205461_b09167_c20130804180545911800_noaa_ops.h5
 \
  All_Data/VIIRS-OCC-EDR_All/Chlorophyll_a clouds.png
-W- colortable cannot be obtained from /rs/lvl2/viirsn/VOCCO/2013/216/
VOCCO_npp_d20130804_t1200057_e1205461_b09167_c20130804180545911800_noaa_ops.h5
 (using default)
VIIRS geo-location samples 3200, lines 3072
```

Figure 14. VIIRS OCC EDR Cloud Flags



Example 15. VIIRS Day-Night-Band Image

In this example, we have a packaged file of the Day-Night Band with its geo-location information. We begin by warping the data to the EastCoast projection using imgMap. And then we make an image.

```
$ imgMap -p All_Data/VIIRS-DNB-SDR_All/Radiance -M ~/aps_v4.98/etc/
default.maps EastCoast test.hdf \
```

```
    GDNBO-
SVDNB_npp_d20120830_t0633359_e0639145_b04354_c20130627134443334345_noaa_ops.h5
$ imgBrowse -c lut=grey -f linear -r 0.0000000003,0.0000000141 test.hdf
 All_Data/VIIRS-DNB-SDR_All/Radiance DNB_npp_d20120830_t0633359.png
```

Figure 15. VIIRS East Coast Day-Night Band

File attributes

The input file is normally an APS file that contains certain attributes and structures required for proper execution of **imgBrowse** program. The following list shows which attributes are used and any alternatives. See APS User's Guide for greater detail.

These attributes are used to annotate the image to describe the type of data that is being displayed and how it was processed.

Attribute	Description
fileTitle	title of file
timeStartYear	Start Year of data
timeStartDay	Start Day of year of data
timeStartTime	Start Time in milliseconds of year of data
timeStart	ASCII start time string (see ctime(3))
timeEndYear	End Year of data
timeEndDay	End Day of year of data
timeEndTime	End Time in milliseconds of year of data
timeEnd	ASCII end time string (see ctime(3))
compType	composite type of data
compMaxPixel	composite maximum pixel
inputMaxPixel	alias for compMaxPixel
sensor	sensor name
Sensor Name	alias for sensor
sensorPlatform	sensor platform
processedVersion	processing version of data
inputMasks	name of masks used in data processing
inputMasksInt	bit-mask of masks used in data processing
mapProjection	name of map projection

These attributes are used to determine how to extract and create the image product.

Attribute	Description
scalingSlope slope scale_factor	calibration scale for conversion of input data to geophysical values

Attribute	Description
Slope	
scalingIntercept intercept add_offset Intercept	calibration offset for conversion of input data to geophysical values
XXXSlope	calibration scale for conversion of input data to geophysical values in XXX units. For example, FahreinheitSlope
XXXIntercept	calibration offset for conversion of input data to geophysical values in XXX units
browseFunc	default scaling function for imgBrowse
browseRange	default data ranges for imgBrowse
validRange	data ranges for imgBrowse if above missing
browseSlope	browse scale for conversion of geophysical values to image
browseIntercept	browse offset for conversion of geophysical values to image
browseCT	default imgBrowse color table number

These attributes describe the MET/OZONE files used by the ocean color programs for MODIS/SeaWiFS processing. If found and the user has select option *-J*, then the MET wind data will automatically be overlayed as vectors and the OZONE data will automatically be overlayed as contours (with no associated colorbars).

Attribute	Description
met1	First input MET file
met2	Second input MET file
met3	Third input MET file
ozone1	First input OZONE file
ozone2	Second input OZONE file
ozone3	Third input OZONE file

Caveats

These are the known problems/bugs with the software.

The maximum image size is limited to 4096 X 4096. This limit is imposed by the off-screen rendering functions of the Mesa 3-D graphics library. This limit can be increased by changing MAX_WIDTH and MAX_HEIGHT in Mesa/src/config.h and recompiling the Mesa library(in fact we have increased these two values from 1280 X 1024), and then recompiling this program with the new library.

This program can not navigate on images which are not warped (like the Level-2 files). Additionally, the warped files must have been created by the program imgMap.

All symbols are drawn in white. There is no option to change these.

Name

imgConvert — convert APS products to another output image format.

Synopsis

`imgConvert` [*`options`*] <*`ifile`*> <*`basename`*> <*`dataset`*> <*`dataset`*>...

Description

By default, the program **imgConvert** will output each data set specified on the command line to its own output file. The output file can be one of: (1) a simple binary file; (2) an ENVI formatted file; or (3) a TIFF image file (with GeoTiFF tags if navigation exists). In some cases, much of the navigation and ancilliary data will not be present in the resulting file as some of these formats are not tuned for this type of information.

For binary files, the image is simply written as native floats. The -I option can be used to output the data set in its original type (usually a 16-bit integer). The output files will be named by appending the name of each data set with the extension `.bin`.

For ENVI output, the result is actually two files. One is a binary dump of the input data converted to floating point. The second file is an ENVI header which describes the data in the binary file. If the input file contains navigation information in the NRL format (see imgMap), then the map projection information is written to the ENVI header. *Note*: The NRL projection software is based on the USGS projection code and contains over 30 different projections. Currently, **imgConvert** will only handle the Mercator map projection. The ENVI format may also be written in multibanded format (see -M). For each single-banded output ENVI file (default), the output file name will consist of the `basename` with the name of each data set product and the extension `.envi`. The ENVI header file will have the extension `.hdr` further appended to it. For a multi-banded ENVI file, the `basename` *will be* the output filename. The ENVI header file in this case will be the `basename` with the extension `.hdr`.

For TIFF files, the image data will be written to a TIFF formatted file. If the input HDF file contains navigation, then GeoTIFF tags will be appended to the TIFF file. The default is to write out the data in floating point format. The -I option may be used to leave the data in its stored format. The output file will be named by appending the name of each data set with the extension `.tiff`.

The user may specify the data sets for output by using regular expressions. These must be put in quotes to protect them from expansion by the shell, however.

Options

-B isp=<isp>, iep=<iep>, isl=<isl>, iel=<iel>, irp=<irp>, irl=<irl>
: defines a subsection of the original image for output.

isp	the starting sample number
iep	the ending sample number
isl	the starting line number
iel	the ending line number
irp	the replication factor along the samples dimension(not implemented)
irl	the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image.

-d	Turns on debugging output.
-I	Do not convert data from original format.
-m <mapFile:mapName>	use the following mapFile and mapName for navigation.
-M	Put all bands into a single file (available only for ENVI files).
-s	Makes GeoTIFF tags standard used-tags. Programs such as Falcon View do not support user-defined tags.
-T <j|d>	Sets the compression option to: JPEG Compression (j) or defalte compression (d)
-v	Verbose mode
--help	Print out a small help guide.
--version	Print out version of software and quit.

Examples

To create an ENVI file of remote sensing reflectances from the file S2000001175134.N3_HNAV:

Example 16. Extracting Remote Sensing Reflectance Products into a single ENVI file

```
$ imgConvert -M S2000001175134.L3_HNAV S2000001175134.envi "rrs_*"
$ ls S2000001175134*
S2000001175134.envi             S2000001175134.envi.hdr
```

To create a GeoTiFF file of diffuse attenuation at 512 nm from the file S2000001175134.N3_HNAV:

Example 17. Extracting Diffuse Attenuation into a GeoTiFF file

```
$ imgConvert S2000079181416.L3_HNAV_GOM S2000079181416.tiff K_532
$ ls S2000079181416*
S2000079181416_K_532.tiff
```

Notes

The binary files produced by imgConvert are in native format. If the file is transferred across platforms, the user will have to handle any and all byte swapping. For example, if writing out a binary file on an SGI and reading the image into MATLAB on a PC. Some formats (like TIFF) and software (like APS) handle the platform conversion automatically.

Name

imgConvolve — compute convolution transformation using a kernel

Synopsis

`imgConvolve` [*options*] <*ifile*> <*ofile*> [<*product*> <*product*>...]

Description

imgConvolve is used to perform a transformation that gives each pixel in an image a new value that is a function of the pixels in its immediate neighborhood. The image to be transformed is in 2-D product array in ifile and the resulting transformed image is stored in an product written to ofile with the same name. The transformation array is user defined and stored in the ofile array as “kernel” *(currently user can only use the two predefined kernels and the code does not store the kernel in the output file)*. Note that if the row number KROW or column number KCOL of the kernel is even, the pixel to the top and left of center is the default central pixel.

The covolution function is:

P'(x,y) = Sum of [K(i,j) * P(x+i,y+j)]

where P is the pixel value at column x and row y, K is the kernel, i ranges from -KROW/2 to KROW/2, and j ranges from -KCOL/2 to KCOL/2. The results are not normalized (see -n option, however).

Options

-k <type> select from a few predefined kernels

```
           1 = Low Pass Filter, 3x3 kernel
               | 1/9, 1/9, 1/9 |
               | 1/9, 1/9, 1/9 |
               | 1/9, 1/9, 1/9 |
           2 = Centre-Weighted Edge Detection, 3x3 kernel
               | -1/8, -1/8, -1/8 |
               | -1/8,  1,   -1/8 |
               | -1/8, -1/8, -1/8 |
           3 = Gaussian Smoothing, 5x5 kernel
               | 1/106, 1/106,  1/106, 1/106, 1/106 |
               | 1/106, 9/106,  9/106, 9/106, 1/106 |
               | 1/106, 9/106, 18/106, 9/106, 1/106 |
               | 1/106, 9/106,  9/106, 9/106, 1/106 |
               | 1/106, 1/106,  1/106, 1/106, 1/106 |
```

-n <N> Normalize the array, that is divide P'(x,y) by N.

-S Sobel edge detection.

-v Forces imgConvolve to run in verbose mode.

--help Print out a small help guide.

--version Print out version of software and quit.

Examples

This call computes a Low Pass Filter on all products in the given file.

Example 18. Applying Low Pass Filter to All Products in a File

```
$imgConvolve -k1 MODPM2004011194000.L3_NOAA_MSB junk
```

This call computes the Center-Weighted Edge Detection on only the remote sensing reflectance products.

Example 19. Applying Center-Weighted Edge Detection to only Reflectance Products in a File

```
$imgConvolve -k2 MODPM2004011194000.L3_NOAA_MSB junk2 'rrs_*'
```

Name

imgDiff — compute difference between product(s) in two files.

Synopsis

`imgDiff` [*`options`*] <*`ifil1`*> <*`ifil2`*> <*`ofile`*> [<*`product`*> <*`product`*>...]

Description

This program computes a simple difference for all products in two input files writing the result to a third file. The user may select the products, which must exist in both input files (though the -A option will allow of alias names to be created), on the command line after the output filename. If no products are given on the command line, then imgDiff will use the file attribute 'prodList to obtain the list of products if it exists.

If the output file is actually one of the input files, then the product name for the difference image will be the product name plus the term “`_diff`”.

The simple difference is comuted as I_1 - I_2. The difference is only calculated for good pixels which are those that have not been masked (e.g., set to LAND/CLDICE) and are not invalid. If the pixel was masked out the result will be zero. If the pixel is invalid in either image, it is considered invalid in the output. The difference is normally written to the output file using the same scaling as the input file. This can be overridden using the -f option.

Besides the simple difference, imgDiff can compute the percent change, percent difference, and the ratio of the two input images. These products will be added to the simple difference (unless the -d option is used). The names of these products will be the product name plus “`_per_chg`”, “`_per_diff`”, and “`_ratio`”.

The percent change is defined as: I_1 - I_2 / I_2

The percent difference is define as:

(1)

The above two products are normally written out as unsigned 16-bit integers with a resolution of 0.01 per scale. Thus, the output contains value in the range from -327.67% to 327.67%. If more precision is required, the -f option will output these as floating point images.

The ratio is defined as I_1 / I_2. If either value is zero, the ratio will be defined as one.

Options

-a	This option allows the user process all common products regardless of type. By default, **imgDiff** will only process arrays marked as “geopyhical products”.
-A name=<alias,...>	This option allows the user to give alias's to products so that differently named products may be differenced. For example, one file may contain `chl_oc3` and the other `chl_oc3m`. Thus the option `-A chl_oc3=chl_oc3m` will then match the `chl_oc3` product from the first file with the `chl_oc3m` product from the second file.
-c	Additionally compute the percent change.

-D	Only write output file if a difference is actually computed. If the two files show no difference this option will cause not output to be generated.
-d	Turn on debugging messages.
-f	For each product, the difference will be written using the same data calibration used as the input file product. This option allows the user to force the differences to be written as floating point.
-k	Count up the number of images that have differences and return this number as an exit status.
-m <min,max>	Use the given range for data filtering when computing the difference. The range is applied to both data sets.
-n	Do not write output file. If this option is used, then the command line should not contain the output file name. Normally used with the -k and -v options.
-o name=<name>, format=<format>, conv=<conv>	Define output file. name — name of the output file format — format of the output file conv — convention of the output file
-p	Additionally compute the percent difference.
-q	Silence all output.
-r	Additionally compute the ratio.
-R	Perform a raw data file comparison. In this case, the two products are compared uncalibrated. For example, if a product is defined as `int16` types in both files, the actual integers are compared. This option precludes any of the other difference products (difference, percent change, and ratio).
-s <slope,int>	This option allows the user to specify the scaling slope and intercept to use for the output product. By default, the scaling of the product in the first file is used. Care must be taken when using this option as it applies to ALL output products.
-S	Generate an Encapsulated PostScript file of the scatter of the difference files.
-t <0|1|2|3|4|5>	This sets the output type and is used in combination with the `-s` option. The values are: (0) for same as type/scaling as input, (1) for 8-bit integer, (2) for 16-bit integer, (3) for 32-bit integer, (4) for floating point, and (5) for double precision.
-x	Turn off masking check.
-v	Forces imgDiff to run in verbose mode.
--help	Print out a small help guide.
--version	Print out version of software and quit.

Examples

In the first example, all the products in the 2.3 file (stored in the file attribure 'prodList') will be subtracted from the 2.4 file with the differences stored in the DIFF file. The product names will be consistant across all files. That is, the product 'rrs_412' found in both the 2.3 and 2.4 files, will yield a product called 'rrs_412' in the DIFF file.

Example 20. Producing a Difference Between Files Created Using Two Versions of APS

```
$imgDiff S2000208182716.L3_HNAV_2.4 S2000208182716.L3_HNAV_2.3 \
         S2000208182716.L3_HNAV_DIFF
```

In this case, we only compute the difference images for the remote sensing reflectances.

Example 21. Producing a Difference Between Files Created Using Two Versions of APS For Only Reflectance Products

```
$imgDiff S2000208182716.L3_HNAV_2.4 S2000208182717.N3_HNAV_2.3 \
         S2000208182716.L3_HNAV_DIFF "^rrs_.*$"
```

Below is an example of using the same output file as input file. Note, that in this case, the output products will be stored back in the 2.4 file with the names: "rrs_412_diff", "rrs_443_diff", etc.

Example 22. Producing a Difference Between Files Created Using Two Versions of APS For Only Reflectance Products (Output Same as an Input)

```
$imgDiff S2000208182716.L3_HNAV_2.4 S2000208182717.N3_HNAV_2.3 \
         S2000208182716.L3_HNAV_2.4 "rrs_*"
```

Below is an example of using the alias option to create a difference product from the chlorophyll-a product from SeaWiFS and MODIS, which use different algorithms and names for the chlorophyll-a product. The `-A` is used to alias the SeaWiFS chorophyll-a product `chl_oc4` to the MODIS chorophyll-a product `chl_oc3`. The resulting output file `chl_diff.hdf` will have the difference in the product `chl_oc4` since that is the name used in the first (SeaWiFS) file.

Note

Both products must be in the list of desired products and that the alias naming is product in first file followed by product in second file.

Example 23. Producing a Difference Between Files Created Using Two Algorithms

```
$imgDiff -A chl_oc4=chl_oc3 seawifs.L3.hdf aqua.L3.hdf chl_diff.hdf chl_oc4
 chl_oc3
```

If the user is only interested in whether there are differences and not in the differences themselfs, then the -k option will be useful. Thus it falls that we are normally not interested in the output file, so the -n option is normally selected. Here is an example, using a shell script:

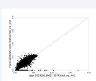
Example 24. Determine Difference Results on a Series of Input Files

```
dir1=/rs/lvl3/seawifs/2.3/MissBight/2001/jan
dir2=/rs/lvl3/seawifs/2.4/MissBight/2001/jan
find $dir1 -type f > /tmp/a.list
find $dir2 -type f > /tmp/b.list
list=`cat /tmp/a.list /tmp/b.list | sort | uniq`
for f in $list
    if imgDiff -kn $dir1/$f $dir2/$f K_532
    then
        echo $f differs
    fi
done
```

Example 25. Scatter Plot

```
file1=aqua.2003295.1022.185713.hdf
file2=meris.2003295.1022.163542.hdf
file3=aqua-meris.2003295.1022.hdf
imgDiff -S $file1 $file2 $file3 rrs_443
```

Figure 16. MODIS-MERIS Rrs@443 Scatter Plot



Name

imgDump — dump data from HDF file

Synopsis

`imgDump` [*`options`*] <*`ifil1`*> <*`ofile`*> <*`prod`*>

Description

This program is used to make ASCII dumps of data from an HDF file. By default, each pixel over land or with a mask value of non-zero (using the default mask of `LAND,CLDICE`) will be printed. The user may select from one of several desired output formats (see *FORMATS*).

Options

-B isp=<isp>, iep=<iep>, isl=<isl>, iel=<iel>, irp=<irp>, irl=<irl>

Defines a subsection of the input image.

isp — the starting sample number

iep — the ending sample number

isl — the starting line number

iel — the ending line number

irp — the replication factor along the samples dimension(not implemented)

irl — the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image.

-B nlat=<nlat>, slat=<slat>, wlon=<wlon>, elon=<elon>, irp=<irp>, irl=<irl>

Defines a geographical subsection of input image.

nlat — the latitude of most North Western point

slat — the latitude of most South Eastern point

wlon — the longitude of most North Western point

elon — the longitude of most South Eastern point

irp — the replication factor along the samples dimension(not implemented)

irl — the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat (see previous -B description).

Selects the desired format. May be either *fox* (default) *ko* or *latlon*.

-F <name>	Name of input mask data set (default *l2_flags*)
-L <file>	Use give file for the land mask file (default `$APS_DATA/landmask.dat`)
-M <names>	A comma separated list of flag names to use for data masking (default is *LAND,CLDICE*).
-r <min,max>	This filters the data by range. Only data falling within these limits will be dumped.
-v	Verbose output.
--help	Print out a small help guide.
--version	Print out version of software and quit.

Formats

The following ASCII formats are available.

FOX

It was originally written to be used as input to Dan Fox's model and therefore output's the Dan Fox File.

152 YYYY MM NNNNN.NN EEEEE.EE DDD HHH SSS.SS	
YYYY	4-digit year
MM	2-digit month
NNNNN.NN	8.2 latitude (degrees North)
EEEEE.EE	8.2 longitude (degrees East)
DDD	3-digit day of month
HHH	3-digit hour of month
SSS.SS	5-digit data value (*e.g.*, sea surface temperature)

KO

It was originally written to be used as input to Ko's model.

YYYY MM DD HH MM SS NNNNN.NN EEEEE.EE DDD HHH SSS.SS	
YYYY	4-digit year
MM	2-digit month
DD	2-digit day of month
HH	2-digit hour of day
MM	2-digit minute of day
SS	2-digit second of day

YYYY MM DD HH MM SS NNNNN.NN EEEEE.EE DDD HHH SSS.SS	
NNNNN.NN	8.2 latitude (degrees North)
EEEEE.EE	8.2 longitude (degrees East)
SSS.SS	5-digit data value (*e.g.*, sea surface temperature)

LATLON

This is a simple format that includes latitude and longitude and data.

NNNNN.NNN EEEEE.EEE SSSSSS.SSSSS	
NNNNN.NNN	9.3 latitude (degrees North)
EEEEE.EEE	9.3 longitude (degrees East)
SSSSS.SSSSS	11.5 digit data value (*e.g.*, sea surface temperature)

Name

imgLandMask — create a land mask product

Synopsis

imgLandMask [*options*] <*ifile*> [<*ofile*>]

Description

This program is used to create a landmask product for the given file. The resulting landmask will be written as 2-D (or 3-D) byte data set with land pixels represented by the value 255 and the water pixels represented by the value 0. If the land and water pixels are given as RGB triplets, then the output data set will have three dimensions. The output data set will be named land_mask.

By default, the input landmask file is `$APS_DATA/landmask.dat` file.

Options

-B isp=<isp>, iep=<iep>, isl=<isl>, iel=<iel>, irp=<irp>, irl=<irl>

Defines a subsection of the input image.

isp	the starting sample number
iep	the ending sample number
isl	the starting line number
iel	the ending line number
irp	the replication factor along the samples dimension(not implemented)
irl	the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image.

-B nlat=<nlat>, slat=<slat>, wlon=<wlon>, elon=<elon>, irp=<irp>, irl=<irl>

Defines a geographical subsection of input image.

nlat	the latitude of most North Western point
slat	the latitude of most South Eastern point
wlon	the longitude of most North Western point
elon	the longitude of most South Eastern point
irp	the replication factor along the samples dimension(not implemented)
irl	the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat (see previous -B description).

-l <c|r,g,b> — Output land pixels using the given value, which must between 0 and 255.

Note

Land and water pixels must have seperate values.

The second option will set the land pixels to the given RGB triplet and produce a 3-D data set.

-L <file> — This option is used to specify the input landmask file.

-n <name> — This option is used to specify another name for the output data set. The default is land_mask

-w <c|r,g,b> — Output 'water' pixels using the given value, which must between 0 and 255.

Note

Land and water pixels must have seperate values.

The second option will set the land pixels to the given RGB triplet and produce a 3-D data set.

--help — Print out a small help guide.

--version — Print out version of software and quit.

Environment Variables

APS_DATA The directory where all the data files exist. Defaults to `$APS_DIR/data`.

Files

`landmask.dat` The landmask file.

`MOD44w.h5` The 250m MODIS Water Mask file.

Name

imgMakeLatLon — create “latitudes” and “longitudes” products for image

Synopsis

`imgMakeLatLon` [*options*] <*ifile*> [<*ofile*>]

Description

This program is used to create a latitude/longitude products for the given input file. The resulting products are stored as 32-bit floating point numbers as decimal degrees (- west, + east). They are the same size as the arrays in the input file and are named latitudes and longitudes by default. This program is useful when latitude/longitude is needed for every point in the image.

Options

-B isp=<isp>, iep=<iep>, isl=<isl>, iel=<iel>, irp=<irp>, irl=<irl>

Defines a subsection of the input image.

- isp — the starting sample number
- iep — the ending sample number
- isl — the starting line number
- iel — the ending line number
- irp — the replication factor along the samples dimension(not implemented)
- irl — the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image.

-B nlat=<nlat>, slat=<slat>, wlon=<wlon>, elon=<elon>, irp=<irp>, irl=<irl>

Defines a geographical subsection of input image.

- nlat — the latitude of most North Western point
- slat — the latitude of most South Eastern point
- wlon — the longitude of most North Western point
- elon — the longitude of most South Eastern point
- irp — the replication factor along the samples dimension(not implemented)
- irl — the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat (see previous -B description).

-l <name>	Rename the latitudes data set to name.
-L <name>	Rename the longitudes data set to name.
-x	This option will write the map coordinates (in meters) to the file. The products will be named mapX and mapY.
--help	Print out a small help guide.
--version	Print out version of software and quit.

Name

imgMap — project satellite images to map projection

Synopsis

`imgMap` [*options*] <*mapName*> <*ofile*> <*ifile*> <*ifile*>...

Description

This program is used to project navigable APS products to a map projection specified by the user. The user should use the program **maps** to create an “image map” that is, an image with a defined number of samples and lines and projection system. Each image map is usually stored in a single file called `default.maps`. This file contains a series of user-defined image maps given a unique name.

The input file must contain either a latitude/longitude product (for each pixel) in the data file or a control points grid. The control point grid is usually defined with the products “CP_Pixels” “CP_Lines” “CP_Latitudes” and “CP_Longitudes.” These are created by default by various programs within APS. Using these points to navigate over the input image (usually in the satellite sensor projection), **imgMap** will fill in the output array from the nearest pixel in the input image. This program does not perform any type of interpolation. If the navigation exists in a separate file, the `-g` may be used to set its location.

This program can handle 2-D and 3-D input arrays. It is assumed that the 3-D data is stored in BIP format.

Options

-B isp=<isp>, iep=<iep>, isl=<isl>, iel=<iel>, irp=<irp>, irl=<irl>

Defines a subsection of the input image.

isp — the starting sample number

iep — the ending sample number

isl — the starting line number

iel — the ending line number

irp — the replication factor along the samples dimension(not implemented)

irl — the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image.

-d

Turn on debugging messages.

-g <file>

Obtain navigation information from <file>. Only useful if a single input file is given.

-G <file>

write out sample/line locations to file

-l lat_name=<name>,lon_name=<name>

set `lat_name` to variable containing latitude data. set `lon_name` to variable containing longitude data

-M <mapFile>	Use the given mapFile rather than the default version.
-o compress=<0|1>	set compress to 1 to compress output file
-v	Forces **imgMap** to run in verbose mode.
--help	Print out a small help guide.
--version	Print out version of software and quit.

Files

`$APS_ETC/default.maps` This is the default mapFile which contains the map provided on the command line.

Environment Variables

$APS_ETC This environmental variable should point to the APS's configuration directory. It is used to find the default maps file. It not set, the -M option can be used to specify the user's map file.

Examples

This example will warp the a_443_qaa and bb_555_qaa products in coms.2013030.0130.021537.D.L2_QAA.goci.00949.v01.500m.hdffile using the JapanSea image map located in the file ~/user/default.maps.

Example 26. Warping Reflectance Data Using Specified Map

```
$ imgMap -M /home/user/default.maps -p a_443_qaa JapanSea \
coms.2013030.0130.021537.D.L3_QAA.goci.JapanSea.v01.500m.hdf
coms.2013030.0130.021537.D.L2_QAA.goci.00949.v01.500m.hdf
```

If $APS_ETC is set then, then user does not have to use the -M option.

Example 27. Warping All Data Using Specified Map From APS Default Maps File

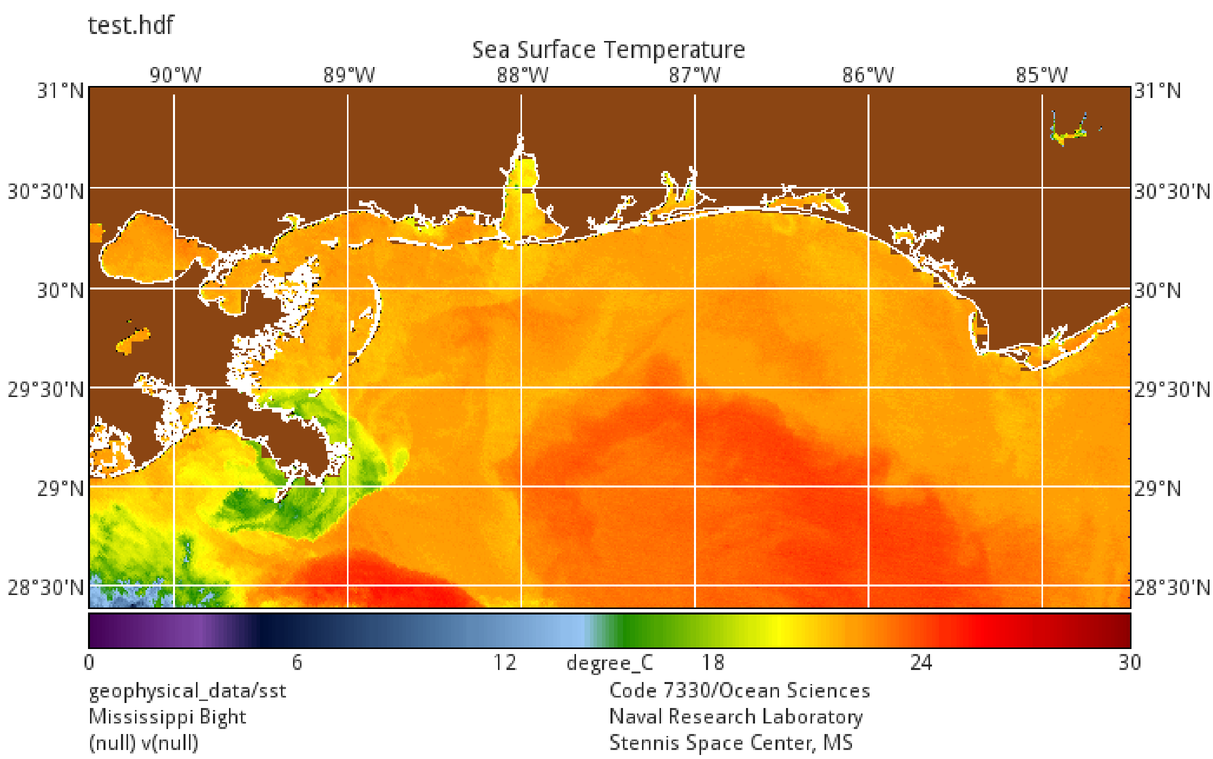
```
$ export APS_ETC=/home/aps/aps_v6.4.4/data
$ imgMap GulfOfMexico viirs_gomex.hdf \
        npp.2013183.0702.172207.D.L2_OC.viirs.08702.v01.750m.hdf \
        npp.2013183.0702.172458.D.L2_OC.viirs.08702.v02.750m.hdf \
        npp.2013183.0702.172333.D.L2_OC.viirs.08702.v01.750m.hdf
```

Example 28. Warping a VIIRS SST file (VSSTO)

```
$ imgMap -M default.maps Gomex gomex.hdf GMTCO-
VSSTO_npp_d20150608_t1827476_e1833280_b18719_c20150714142044089607_noaa_ops.h5
```

Example 29. Warping a MODIS SST file from NASA (netCDF version)

```
$ export APS_ETC=/home/aps/aps_v6.4.4/data
$ imgMap -p geophysical_data/sst -I lat_name=navigation_data/
latitude,lon_name=navigation_data/longitude MissBight
 A2013134073500.L3.MissBight.hdf A2013134073500.L2_LAC_SST.nc
processing A2013134073500.L2_LAC_SST.nc
-W- unable to copy src history to dst history (status=1)
$ imgBrowse -r 0,30 test.hdf geophysical_data/sst
 A2013134073500.L3.MissBight.sst.png
-W- colortable cannot be obtained from test.hdf (using default)
-W- unable to open flags l2_flags
```



Name

imgMean — calculate Mean/Min/Max/StDev of a series of images.

Synopsis

`imgMean` [*`options`*] <*`ifil1`*> <*`ifil2`*>...

Description

This program will produce an image of the mean for each pixel for each selected product (`-H`) in a series of input files. The production of the mean may be filtered by the use of data masks (`-m`, `-t`, and `-z`) or input data ranges (`-r`) with applicable functions applied (`-f`). The program can also produce the minimum (`-n`), maximum (`-x`), counts (`-c`), and standard deviation (`-s`) of these products. Currently the program is limited by the operating system with respect to the maximum number of input files. A shell script has been created that when used with the `-F` option can work around this limit. However, some capabilities are unavailable in these cases.

The output mean will be named the same as the input product. The others (minimum, maximum, standard deviation, and count) will be the concatenation of the input product name with an appropriate extension. For example, if the input product is `a_443_qaa`, then the minumum will be written to `a_443_qaa_min`. Likewise, the maximum will be written to `a_443_qaa_max` product. The count and standard deviation extensions will are: `_cnt` and `_stdev`.

In addition, this program can make incremental composites (the addition of one or more files to an already created composite file) and can merge two or more composite files. See "Incremental Composite" below for more details.

For `l2_flags` products, the compositing is done as follows. Until a compositable pixel is found, l2_flags are OR'ed. should no compositable pixels be encountered, the resulting l2_flag is the OR of all of the l2_flags of the images.

Once a compositable pixel has been found, the l2_flags are AND'ed with other compositable pixels.

Lastly, the **imgMean** can product a (weighted) “latest” pixel composite. See the section below on this special type of composite.

Options

-a	Do not create the average (mean) image.
-c	Output a "count" product.
-C <product>	The name of the product containing the cloud albedo for each pixel. Default: "cloud_albedo".
-F <filename>	Get files to composite from "filename". The format of this file is one file name per line, the whole line is used, and may contain spaces or any other character. DO NOT QUOTE FILENAMES. Any combination of Level-3 and NRL Level-4 files may be used.
-f <n>	Apply the given function to the input data before determining any statistics with the data. The number represents the available functions which are: 0 for none, 1 for log10, 2 for alog10, 3 for ln, 4 for exp. The function is applied after any range checks are performed. Multiple functions may be specified by separating the functions by a

comma. If fewer functions are specified than products, no function will be used with the remaining products. The -f option is only valid when -H is also specified.

```
-H Kd_488_lee,chlor_a -f 0,1
```

Kd_488_lee	function none
chlor_a	function log10

or

```
-H Kd_488_lee,chlor_a,bb_555_qaa -f 0,1
```

Kd_488_lee	function none
chlor_a	function log10
bb_555_qaa	function none

-H <product> — Designates that the files to be used in making composites will be in HDF format. The user must supply the name of which product to use. Multiple product's may be specified by separating them with comments.

```
-H Kd_488_lee,chlor_a
```

-I # — Used to specify the value to be used in replacing invalid data.

-k — Compute a count product for each flag that computes the sum total of the number of times a particular flag is set. The product is named after the flag name (for example, LAND or CLDICE).

-l — Composite the l2_flags.

-L — Do an incremental composite. See "Incremental Composite" below.

-m <mask> — The mask value used to filter out pixels from the compositing. The mask may be specified as an integer or as a comma separated string of flag names.

ex. To mask out the ATMFAIL and LAND flags, use either of the following: `-m 3` or `-m ATMFAIL,LAND`

-M product — The name of the mask array to use for masking. The default is "l2_flags".

-n — Produces an image of the minimum value of all input images at each corresponding pixel location.

-o <outfile> — Set the output filename to `<outfile>`. Without this option, an output filename will be created based on the start and end times of the input files. The name will be of the form: `platform.YYYYJJJ.MMDD.YYYYJJJ.MMDD.D.L4_TT.region.sensor.version.resolu` where the first group of YYYYJJJ is the earliest start year and julian day and the second group is the latest end year and julian day, the TT is the composite type as set by the -T option. TT will be WE for weekly, MO for monthly, YR for yearly, and RL for Rolling composites. This follows the standard APS naming scheme and is available for such inputs. For other inputs, this scheme may insert `unknown`.

-p <n> — Define the number of valid pixels to use. Default value is 1. This option essentially creates what is known as the latest pixel composite since the first valid pixel is retained. For proper implmentation, the files should be provided on command line or through a list file in reverse chronical order.

-P <d|i>	Produces an image of the latency of all input images at each corresponding pixel location. This product replaces the count image in latest pixel composites and forces a maximum valid pixel value of 1 (*i.e.*, -p 1). When the *-P i* option is used, the product will contain the file index (1-relative) of the file used in the latest pixel composite. The file used can be found in the *inputFiles* file attribute. For pixels that are invalid, a index of zero will be written. When the *-P d* option is used, the pixel will contain the number of days from the most recent file. Thus, a value of 0 will indicate pixels that are the most recent. A value of 7 will indicate pixels that are seven days old. Invalid pixels will contain USHRT_MAX.

-r <lower,upper>	Set the lower and upper bounds for range checking. A lower and upper bounds may be specified and must be separated by a comma. Ranges for multiple products may be specified by separating the ranges by a colon. The -r option is only valid when the -H option is also specified.
	
	`-H K_532 -r .01,6` or `-H K_532,chlor_a -r .01,6:.01,64`
	
	If fewer ranges are specified using -r than products using -H then the last value in the -r list will be for the rest of the -H products.
	
	`-H K_532,chlor_a, chl_stumpf,... -r .1,5:.1,64`
	
	K_532 — range .1 - 5
	chlor_a — range .1 - 64
	chl_stumpf — range .1 - 64
	... — range .1 - 64
	
	If no ranges are specified, then the ranges from the 'validRange' attribute are used for range checking.

-s	Produces an image of the standard deviation for each corresponding pixel for all input images.
-t albedo	This defines the albedo value to use when compositing data. To be used the input files must contain the *cloud_albedo* product.
-T <type>	Specify type of composite. 1 for daily, 2 for weekly, 3 for monthly, 4 for yearly, 5 for latest pixel composite, 6 for rolling composites and 7 for seasonal. This is mainly an informational option to describe the composite. It adds a suite for attributes including *compType*, *compStartTimeFrame*, *compEndTimeFrame*, *compTimeFrame*.
-v	Forces imgMean to run in verbose mode.
-W <weights>	Creates a weighted average based upon the weight table given in weights. The output product for the weighted average will be "_weight". *NOTE* Currently incremental composites are not possible with this option. weights is defined as a string of comma "," seperated floating point numbers which define your weight table. `-W .6,.2,.1,.05,.04,.01` would define a weight table as follows [all valid pixels in the first input file] * .6 [all valid pixels in the second input file] * .2 [all valid pixels in the third input file] * .1 etc.

The sum of the valid pixels * their respective weights is divided by the addition of those weights that were used for each pixel location.

5 files are input, using a weight table of .5,.4,.3,.2,.1 for pixel location p files 1,2, and 5 have valid data, the weighted mean would be calculated as follows: ([pixel 1] * .5 + [pixel 2] * .4 + [pixel 5] * .1) / (.5 + .4 + .1)

-w <start,stop> — Define the week start and stop times for the *-T2* composites. Used when creating *compStartFrame*/*compEndFrame* attributes.

-x — Produces an image of the maximum value of all input images at each corresponding pixel location.

-z <threshold> — Set the sensor zenith angle threshold. imgMean will ignore pixels whose sensor zenith angle(degrees) is above this threshold.

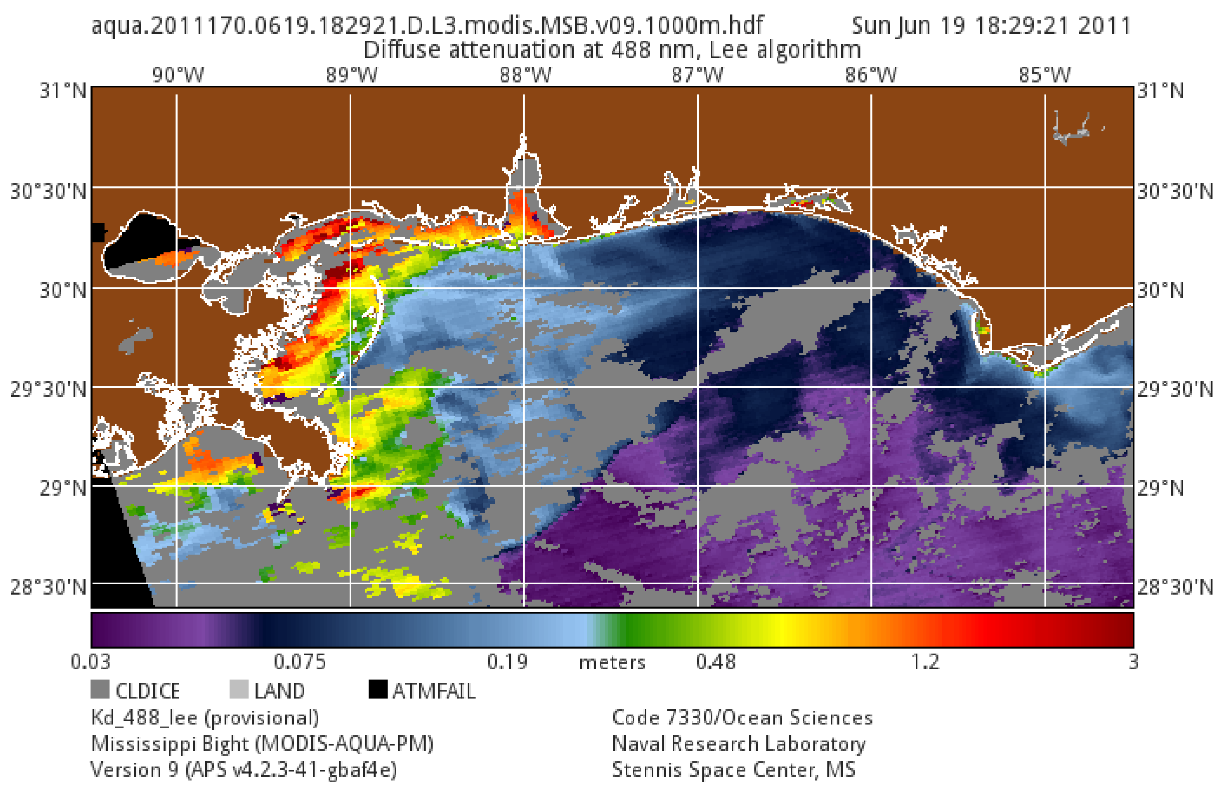
-Z <product> — The name of the product containing the sensor zenith angles(degrees) for each pixel. Default: "senz".

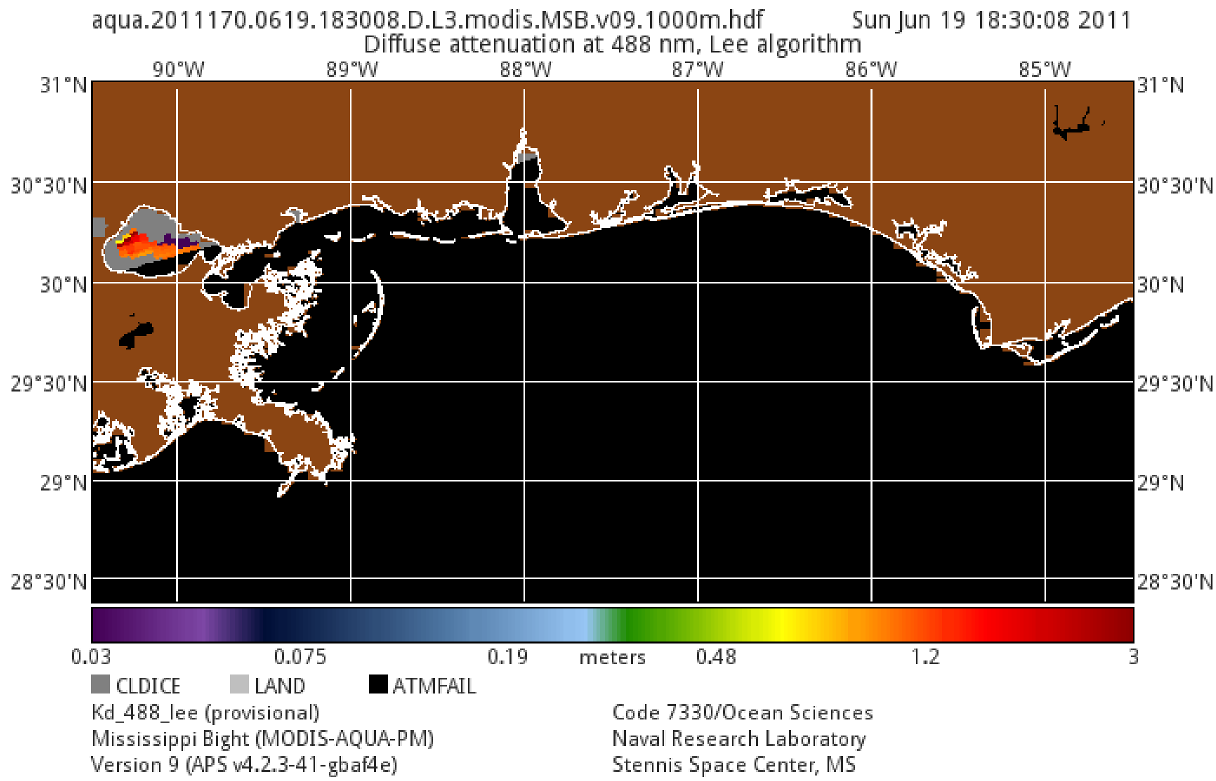
--help — Print out a small help guide.

--version — Print out version of software and quit.

Simple Mean

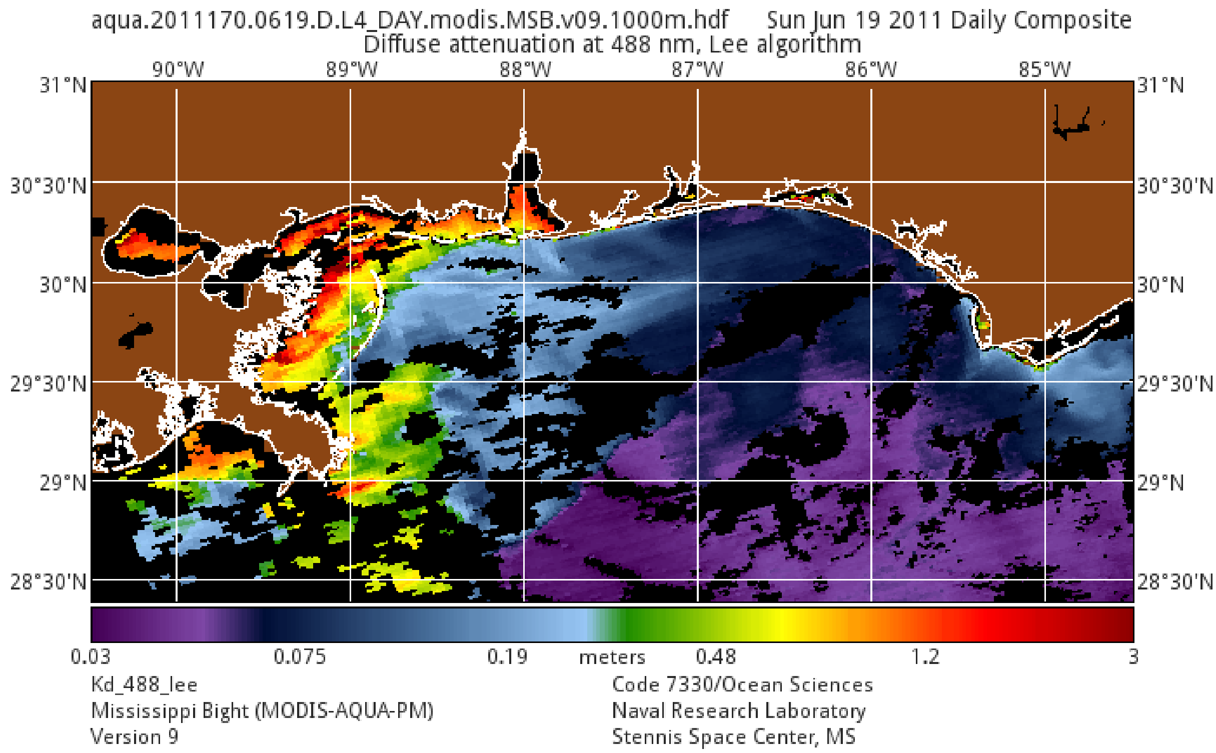
In this example, a daily composite is created by averaging two individual files for the given day. The MODIS pass splits Lake Ponchartarain in half as can be seen in these two images.






```
$ file1=$srcdir/aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf
$ file2=$srcdir/aqua.2011170.0619.183008.D.L3.modis.MSB.v09.1000m.hdf
$ ofile=aqua.2011170.0619.D.L4_DAY.modis.MSB.v09.1000m.hdf
$ rm -f $ofile
$ imgMean -vv -T 1 -H Kd_488_lee,l2_flags -o $ofile $file1 $file2
```

The final image.



Here are some dumps of the data about a point Lake Pontchartrain around the location of the split. The points marked `invalid` correspond to the black parts of the image. These are where no data exists. The values `-6.5534` correspond to the locations where the Level-2 product failed to be generated. This could be due to cloud cover, glint, haze, or other issue with the product generation.

Kd_488_lee : aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf

```
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid  1.1562  1.1562  1.1562  1.1254
invalid invalid invalid  1.1562  1.1562  1.1562  1.1562  1.0608  1.0752
invalid  1.0466  1.0466  0.9794  1.0608  1.0608  1.0608  1.0608  1.0752
 0.9794  0.9794  0.9794  0.9794  1.0608  1.0608  1.0608  1.0130  1.0130
 0.9794  0.9794  0.9794  0.9818  1.0130  1.0130  1.0130  1.0130  1.0130
 0.9818  0.9818  0.9818  0.9818  0.9818  1.0130  1.0130  1.0848  1.0848
 0.9818  0.9818  0.9818  0.9818 -6.5534  1.0848  1.0848  1.0848  1.0848
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534  1.0848  1.0848 -6.5534 -6.5534
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534
```

Kd_488_lee : aqua.2011170.0619.183008.D.L3.modis.MSB.v09.1000m.hdf

```
1.1626 -6.5534 -6.5534 -6.5534  1.2152  1.2152  1.2152 -6.5534 -6.5534
 1.1168  1.1168  1.2152  1.2152  1.2152  1.1378  1.1378  1.1088  1.1088
 1.1168  1.1168  1.0650  1.1378  1.1378  1.1378  1.1378  1.1088  1.1088
 1.0650  1.0650  1.0650  1.1378  1.1378  1.0776  1.0776  1.0776  1.0890
```

```
 1.0650  1.0650  1.0490  1.0776  1.0776  1.0776  1.0776  1.0776  1.0890
 1.0490  1.0490  1.0490  1.0490  1.0776  1.0776  1.0216  1.0216  1.0540
 1.0490  1.0490  0.9970  0.9970  1.0216  1.0216  1.0216  1.0216  1.0540
 0.9970  0.9970  0.9970  0.9970  1.0216  1.0216  1.0780  1.0780  1.0780
 0.9970  0.9970 -6.5534 -6.5534  1.0780  1.0780  1.0780 invalid invalid
-6.5534 -6.5534 -6.5534 invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
```

Kd_488_lee : aqua.2011170.0619.D.L4_DAY.modis.MSB.v09.1000m.hdf

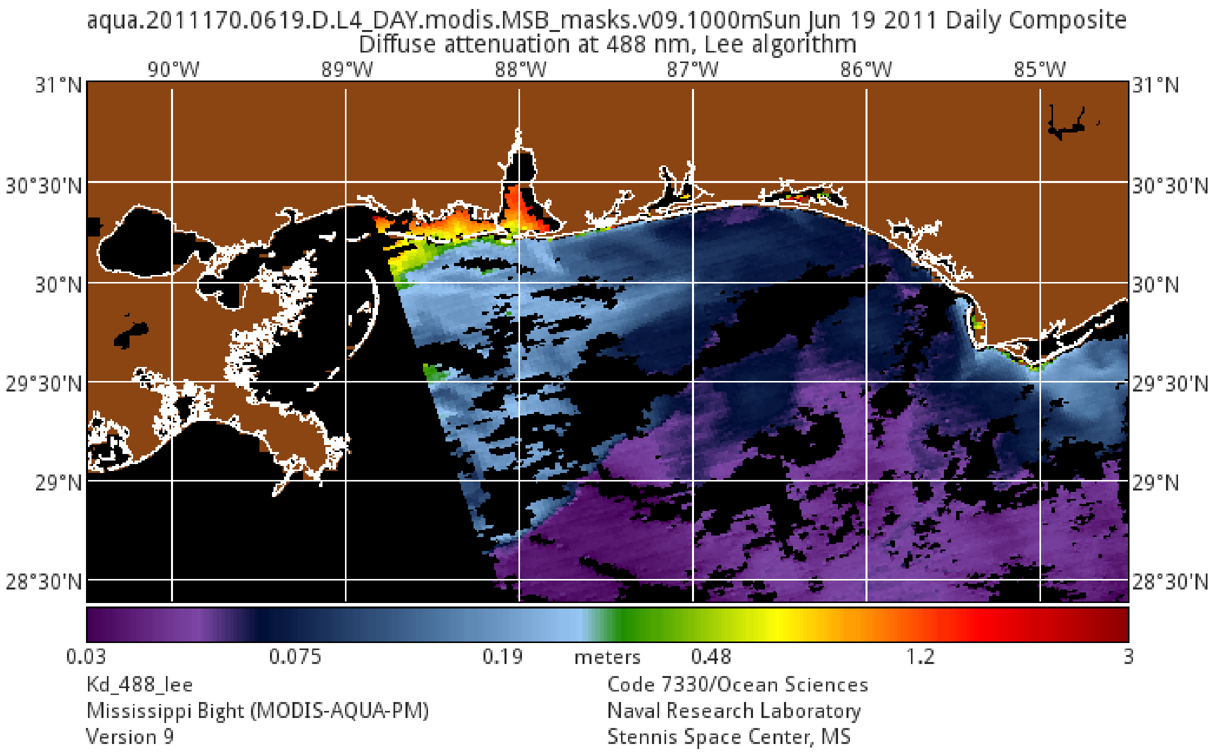
```
1.1626 invalid invalid invalid  1.2152  1.2152  1.2152 invalid invalid
 1.1168  1.1168  1.2152  1.2152  1.2152  1.1378  1.1378  1.1088  1.1088
 1.1168  1.1168  1.0650  1.1378  1.1378  1.1470  1.1470  1.1326  1.1172
 1.0650  1.0650  1.0650  1.1470  1.1470  1.1170  1.1170  1.0692  1.0822
 1.0650  1.0558  1.0478  1.0286  1.0692  1.0692  1.0692  1.0692  1.0822
 1.0142  1.0142  1.0142  1.0142  1.0692  1.0692  1.0412  1.0174  1.0336
 1.0142  1.0142  0.9882  0.9894  1.0174  1.0174  1.0174  1.0174  1.0336
 0.9894  0.9894  0.9894  0.9894  1.0018  1.0174  1.0456  1.0814  1.0814
 0.9894  0.9894  0.9818  0.9818  1.0780  1.0814  1.0814  1.0848  1.0848
invalid invalid invalid invalid invalid  1.0848  1.0848 invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
```

In first line, notice that the daily composite contains `invalid` for the bad data in the second pass (represented by `-6.5534`). In the third line, notice that the first five values are exactly the same as the first five values in the same line of the second pass. This is due to the `invalid` values in the first pass. The remaining four values of the third line are, however, the average of both passes.

Effect of Data Masking on Composite Results

Now, if we added the following processing masks, we will get the following results.

```
$ rm -f $ofile
$ imgMean -vv -m "NAVFAIL,HISATZEN,HISOLZEN" -T 1 -H Kd_488_lee,l2_flags -o
 $ofile $file1 $file2
```

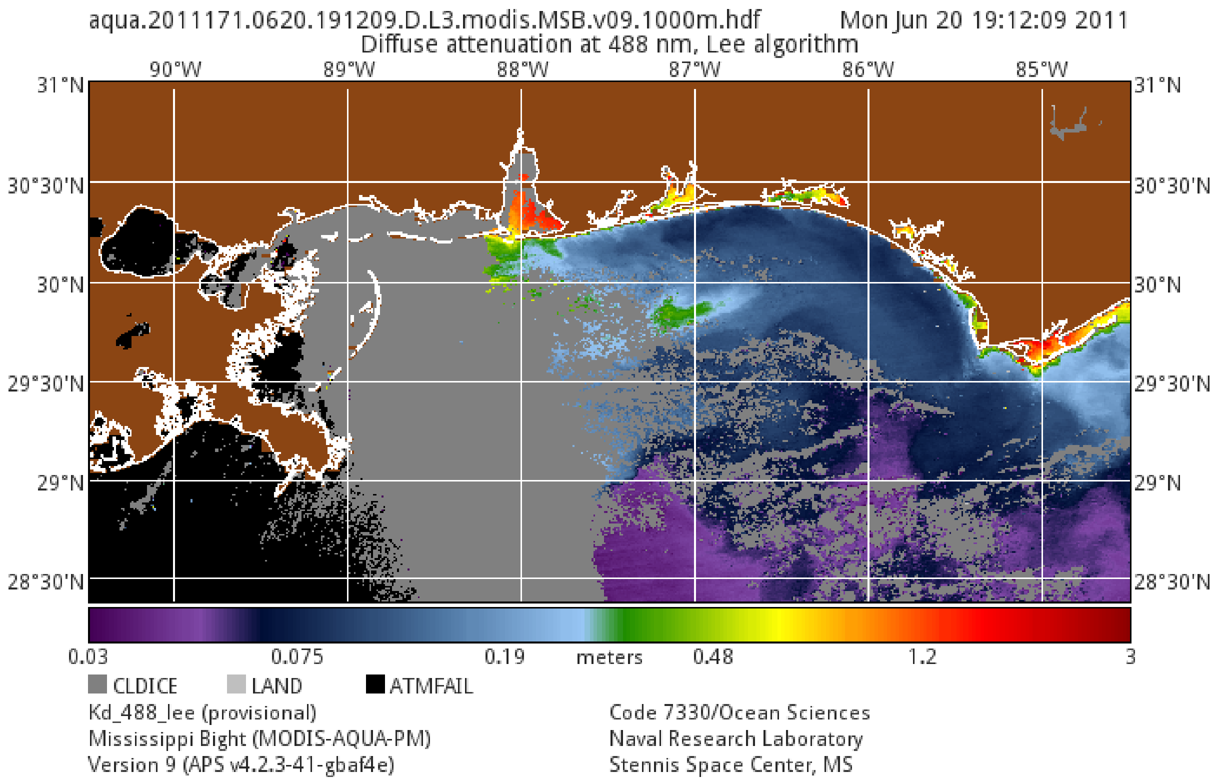


An image dump of the Lake Ponchartrain location gives us no results since it was masked.

```
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
```

Expanding on the Simple Average

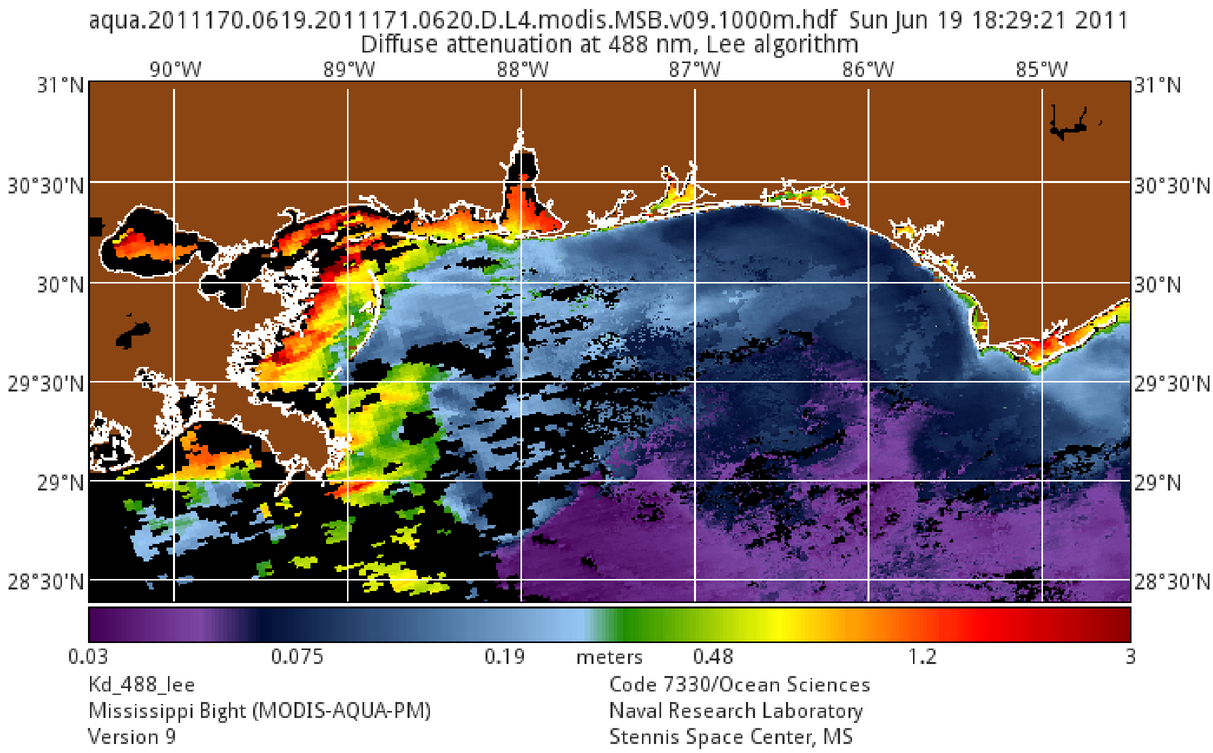
This image is MODIS pass for the very next day as those used by the previous two sections. We demonstrate the effect of a composite all three files that cover two days of satellite colletion.



Now to produce an 2-day average using all three files, we use the following command.

```
$ file1=$srcdir/aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf
$ file2=$srcdir/aqua.2011170.0619.183008.D.L3.modis.MSB.v09.1000m.hdf
$ file3=$srcdir/aqua.2011171.0620.191209.D.L3.modis.MSB.v09.1000m.hdf
$ ofile=aqua.2011170.0619.2011171.0620.D.L4.modis.MSB.v09.1000m.hdf
$ rm -f $ofile
$ imgMean -vv -T 1 -H Kd_488_lee,l2_flags -o $ofile $file1 $file2 $file3
```

This is our 2-day composite



Maximum Number of Files

There are limits imposed by the operating system for the total number of open files by a process. This limit is obtained by **imgMean**. You can determine this limit at the shell using **ulimiit -n** To obain a composite of more files than this limit, a shell script version of this program **imgMean.sh** is provided to over-come it.

The script will call **imgMean** successfully using the incremental composite capability in batches of 200 files each until all the files are completed. This is done by creating a series of temporary files. These files are of the form: `.imgMean_XXXXX`. If **imgMean** aborts, this hidden files may remain on the filesystem.

Incremental Composites

The term "incremental composite" is used to refer to building a long range composite by successively adding data to a "running" file. For example, to create a monthly composite in a near real-time system, we begin with data from the 1st day of the month and compute a composite. This composite becomes our initial monthly composite.

When we obtain the 2nd day of data, we take the inital monthly composite and add the data from the 2nd day. That is saved as our new monthly composite. On day three, we append to the monthly composite all the data collected on the 3rd day. We continue this until the last day of the month, when the computed monthly composite is complete. The next day's processing will start the next months composite.

Now to accomplish this, our running monthly composite must contain the mean and the number of pixels (the count product) that made up the mean. [1] To produce the count, the `-c` option must be used. To have **imgMean** read the count for the next composite, the `-L` option must be used.

```
$ file1=aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf
```

[1] As well as the sum of the squares if we plan to produce a monthly standard deviation product.

```
$ file2=aqua.2011170.0619.183008.D.L3.modis.MSB.v09.1000m.hdf
$ ofile=running_composite.hdf
$ rm -f $ofile
$ imgMean -vv -T 4 -c -H Kd_488_lee,l2_flags -o $ofile $file1 $file2

$ file3=aqua.2011171.0620.191209.D.L3.modis.MSB.v09.1000m.hdf
$ mv $ofile temp.hdf
$ imgMean -vv -T 4 -Lc -H Kd_488_lee,l2_flags -o $ofile temp_hdf $file3

$ file4=aqua.2011172.0621.181706.D.L3.modis.MSB.v09.1000m.hdf
$ file5=aqua.2011172.0621.195511.D.L3.modis.MSB.v09.1000m.hdf
$ mv $ofile temp.hdf
$ imgMean -vv -T 4 -Lc -H Kd_488_lee,l2_flags -o $ofile temp_hdf $file4
$file5

$ file6=aqua.2011173.0622.185951.D.L3.modis.MSB.v09.1000m.hdf
$ mv $ofile temp.hdf
$ imgMean -vv -T 4 -Lc -H Kd_488_lee,l2_flags -o $ofile temp_hdf $file5
```

Most Recent (Latest Pixel) Composites

The term “latest pixel composite” is used to refer to the building of a long range composite by extracting the first valid pixel(s) from a series of input images. The number of pixels considered is controlled by the `-p` option. When set to one, the resulting composite will consist of the first valid pixel from the input pixel. When the input files are ordered in reverse time order (newest to oldest), then the composite is termed “latest pixel composite”. Note that, the user must provided those two conditions for such a composite.

When considering more than the first valid pixel (`-p n` such that `n > 1`), the resulting composites might also have a weight applied to each successive pixel. For example, when considering two pixels, the weights might be set to produce an output in which the first pixel is considered more than the second pixel.

Consider the following files given in the order below and the most recent pixel option `-p` set to 1. Some pixel dumps are provided for each file. The index is the position of that file in this file starting with 1. The day is the number of days since the *most recent* day of this set. The last file contains data from the 24th of June 2011. The first file contains data from *five* days prior.

Note

The example here has the data listed from *oldest* to *newest*. This order is backwards if one wants a true latest pixel composite. It is listed this way here to emphasis the way the most recent pixel algorithm works.

```
$ cat rolling.lst
aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011170.0619.183008.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011171.0620.191209.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011172.0621.181706.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011172.0621.195511.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011173.0622.185951.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011175.0624.184739.D.L3.modis.MSB.v09.1000m.hdf
$ ofile=aqua.2011170.0619.2011175.0624.D.L4_LP_index.modis.MSB.v09.1000m.hdf
```

```
$ rm -f $ofile
$ imgMean -F rolling.lst -p -T 6 -H chlor_a,Kd_488_lee,l2_flags -f 1,0,0 -o
$ofile
```

Kd_488_lee : aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf index = 1, day = 5

```
$ imgRead aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534  0.0478  0.0478
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534  0.0478  0.0498
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534  0.0498  0.0498
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534  0.0520
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534  0.0468  0.0468
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534  0.0468  0.0468
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534  0.0460  0.0460
```

Kd_488_lee : aqua.2011170.0619.183008.D.L3.modis.MSB.v09.1000m.hdf index = 2, day = 5

```
$ imgRead aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
invalid invalid invalid invalid invalid invalid invalid invalid invalid
```

Kd_488_lee : aqua.2011171.0620.191209.D.L3.modis.MSB.v09.1000m.hdf index = 3, day = 4

```
$ imgRead aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf
-6.5534  0.0780  0.0674 -6.5534  0.0660  0.0662  0.0670  0.0670  0.0640
-6.5534 -6.5534  0.0562  0.0448  0.0674  0.0620  0.0604  0.0636  0.0654
 0.0586  0.0682  0.0694  0.0660  0.0672  0.0662  0.0598  0.0654  0.0654
 0.0730  0.0682  0.0694  0.0692  0.0690  0.0688 -6.5534 -6.5534  0.0638
-6.5534  0.0768  0.0678  0.0692  0.0716  0.0654  0.0658  0.0658  0.0652
 0.0606  0.0574  0.0668  0.0692  0.0690  0.0664  0.0664  0.0698  0.0654
 0.0652  0.0694  0.0668  0.0682  0.0690 -6.5534  0.0718 -6.5534  0.0694
 0.0670  0.0682  0.0694  0.0810  0.0810 -6.5534  0.0640 -6.5534 -6.5534
 0.0914  0.0710  0.0686  0.0686  0.0636 -6.5534 -6.5534 -6.5534 -6.5534
```

```
-6.5534 -6.5534 -6.5534  0.0744 -6.5534 -6.5534 -6.5534 -6.5534  0.0658
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534  0.0520  0.0520
```

Kd_488_lee : aqua.2011172.0621.181706.D.L3.modis.MSB.v09.1000m.hdf index = 4, day = 3

```
$ imgRead aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf
 0.0462 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534
 0.0484  0.0484 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534
 0.0484  0.0484 -6.5534  0.0652  0.0652  0.0652  0.0652 -6.5534 -6.5534
 0.0574  0.0574  0.0652  0.0652  0.0652  0.0652  0.0652  0.0536  0.0536
 0.0574  0.0574  0.0574 -6.5534 -6.5534 -6.5534 -6.5534  0.0536  0.0536
 0.0498  0.0498  0.0498 -6.5534 -6.5534 -6.5534 -6.5534  0.0576  0.0576
 0.0498  0.0498  0.0498  0.0664  0.0664  0.0664  0.0664  0.0664  0.0576
 0.0570  0.0570  0.0570  0.0664  0.0664  0.0664  0.0664 -6.5534  0.0508
 0.0570  0.0570  0.0570  0.0570 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534
 0.0702  0.0702 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534 -6.5534
-6.5534 -6.5534 -6.5534 -6.5534 -6.5534  0.0510  0.0510  0.0510  0.0510
```

Here is the result of the composite. Notice that the most recent valid pixel is the result. The first file contains bad data (represented by the `-6.5534`) which are skipped. The second file contains only invalid data. The third file contains both bad and good data. For each pixel where the first image was bad, but the third image was good, the good pixel is used. Again this is repeated for the fourth file.

```
$ imgRead aqua.2011170.0619.2011175.0624.D.L4_LP.modis.MSB.v09.1000m.hdf
 0.0462  0.0780  0.0674 invalid  0.0660  0.0662  0.0670  0.0670  0.0640
 0.0484  0.0484  0.0562  0.0448  0.0674  0.0620  0.0604  0.0478  0.0478
 0.0586  0.0682  0.0694  0.0660  0.0672  0.0662  0.0598  0.0478  0.0498
 0.0730  0.0682  0.0694  0.0692  0.0690  0.0688  0.0652  0.0498  0.0498
 0.0574  0.0768  0.0678  0.0692  0.0716  0.0654  0.0658  0.0658  0.0652
 0.0606  0.0574  0.0668  0.0692  0.0690  0.0664  0.0664  0.0698  0.0654
 0.0652  0.0694  0.0668  0.0682  0.0690  0.0664  0.0718  0.0664  0.0694
 0.0670  0.0682  0.0694  0.0810  0.0810  0.0664  0.0640 invalid  0.0520
 0.0914  0.0710  0.0686  0.0686  0.0636 invalid invalid  0.0468  0.0468
 0.0702  0.0702 invalid  0.0744 invalid invalid invalid  0.0468  0.0468
invalid invalid invalid invalid invalid  0.0510  0.0510  0.0460  0.0460
```

The second dump shows the index as we have just described. An index of zero means that no valid pixel was found from any of the input files. This output is obtained using the `-Pi` option. It is stored in a “latency” product. The name is generated by the product name and `latency` The following dump is from `Kd_488_lee_latency`.

```
$ imgRead
aqua.2011170.0619.2011175.0624.D.L4_LP_index.modis.MSB.v09.1000m.hdf
 4.0000  3.0000  3.0000  0.0000  3.0000  3.0000  3.0000  3.0000  3.0000
 4.0000  4.0000  3.0000  3.0000  3.0000  3.0000  3.0000  1.0000  1.0000
 3.0000  3.0000  3.0000  3.0000  3.0000  3.0000  3.0000  1.0000  1.0000
```

```
 3.0000  3.0000  3.0000  3.0000  3.0000  3.0000  4.0000  1.0000  1.0000
 4.0000  3.0000  3.0000  3.0000  3.0000  3.0000  3.0000  3.0000  3.0000
 3.0000  3.0000  3.0000  3.0000  3.0000  3.0000  3.0000  3.0000  3.0000
 3.0000  3.0000  3.0000  3.0000  3.0000  4.0000  3.0000  4.0000  3.0000
 3.0000  3.0000  3.0000  3.0000  3.0000  4.0000  3.0000  0.0000  1.0000
 3.0000  3.0000  3.0000  3.0000  3.0000  0.0000  0.0000  1.0000  1.0000
 4.0000  4.0000  0.0000  3.0000  0.0000  0.0000  0.0000  1.0000  1.0000
 0.0000  0.0000  0.0000  0.0000  0.0000  4.0000  4.0000  1.0000  1.0000
```

This last dump shows the latency product (`Kd_488_lee_latency`) when using day as output (`-Pd`). In this case, the value is set to the number of days since the most recent day in the list of input files.

Note

As noted above, this list is in the *wrong* order for the so called “latest pixel composite”. It is up to the user to insure that the inputs are ordered properly.

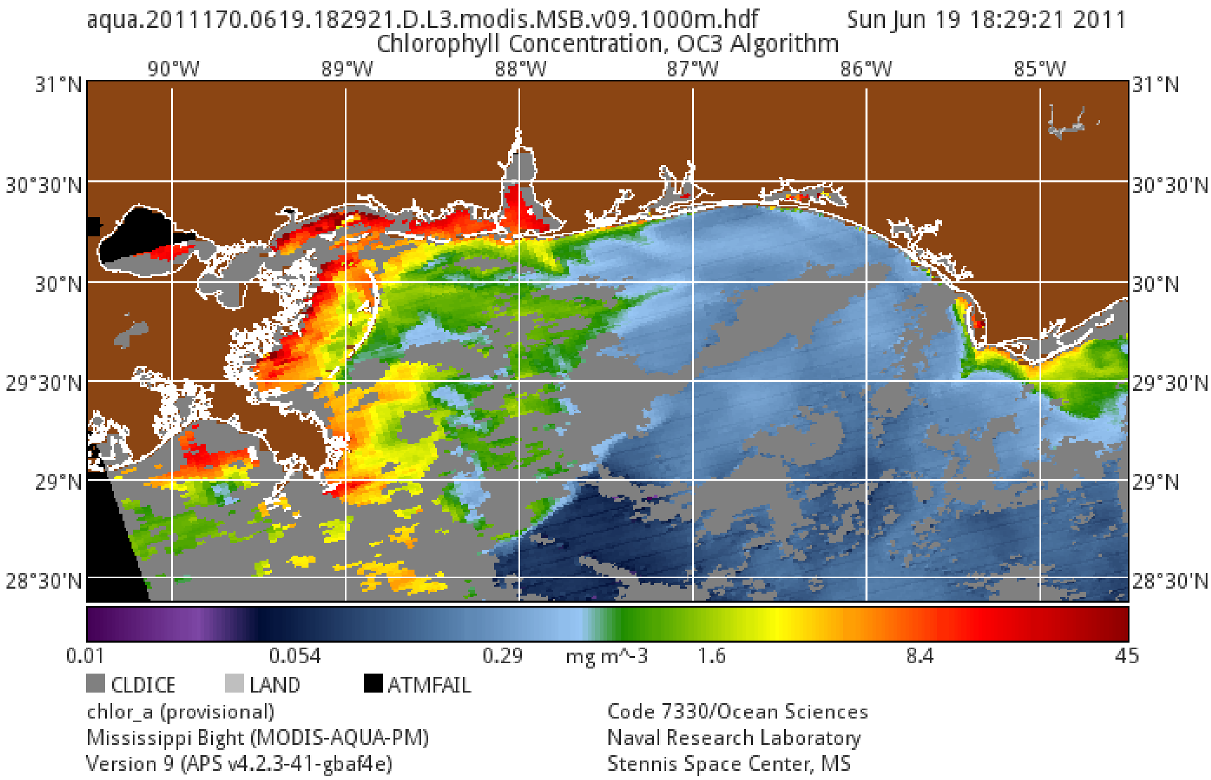
```
$ imgRead aqua.2011170.0619.2011175.0624.D.L4_LP_day.modis.MSB.v09.1000m.hdf
 3.0000  4.0000  4.0000 invalid  4.0000  4.0000  4.0000  4.0000  4.0000
 3.0000  3.0000  4.0000  4.0000  4.0000  4.0000  4.0000  5.0000  5.0000
 4.0000  4.0000  4.0000  4.0000  4.0000  4.0000  4.0000  5.0000  5.0000
 4.0000  4.0000  4.0000  4.0000  4.0000  4.0000  3.0000  5.0000  5.0000
 3.0000  4.0000  4.0000  4.0000  4.0000  4.0000  4.0000  4.0000  4.0000
 4.0000  4.0000  4.0000  4.0000  4.0000  4.0000  4.0000  4.0000  4.0000
 4.0000  4.0000  4.0000  4.0000  4.0000  3.0000  4.0000  3.0000  4.0000
 4.0000  4.0000  4.0000  4.0000  4.0000  3.0000  4.0000 invalid  5.0000
 4.0000  4.0000  4.0000  4.0000  4.0000 invalid invalid  5.0000  5.0000
 3.0000  3.0000 invalid  4.0000 invalid invalid invalid  5.0000  5.0000
invalid invalid invalid invalid invalid  3.0000  3.0000  5.0000  5.0000
```

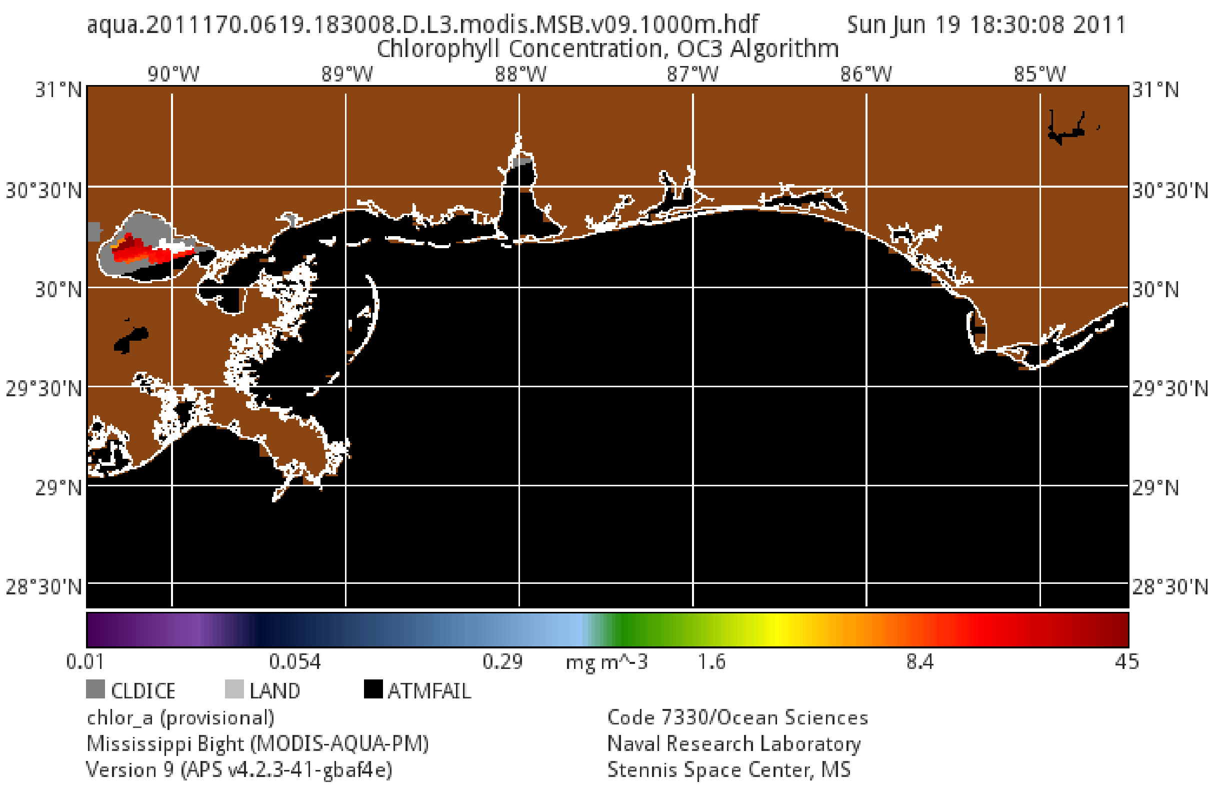
A “Rolling” Composite Mean

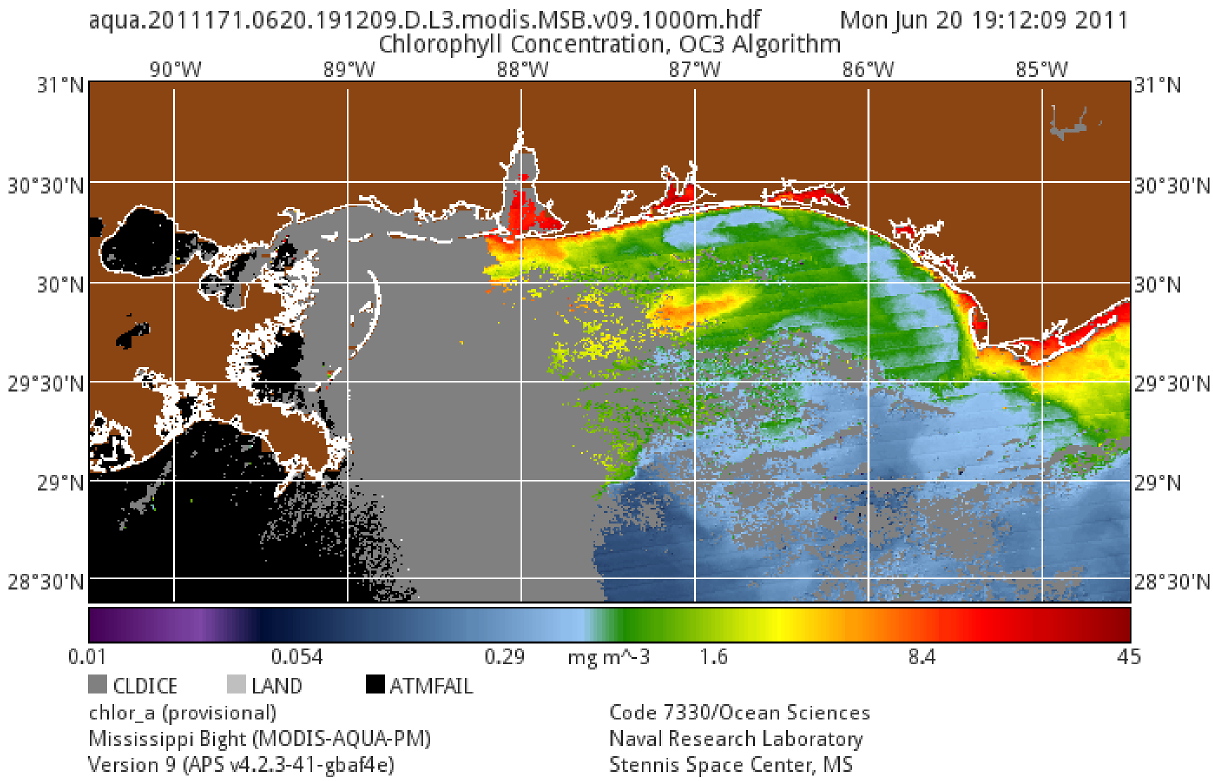
Now to produce an 6-day “rolling” average, use the following command. In this case, we use the `-F` option to select our file list and since we are compositing the chlor_a product, we add the `-f` option.

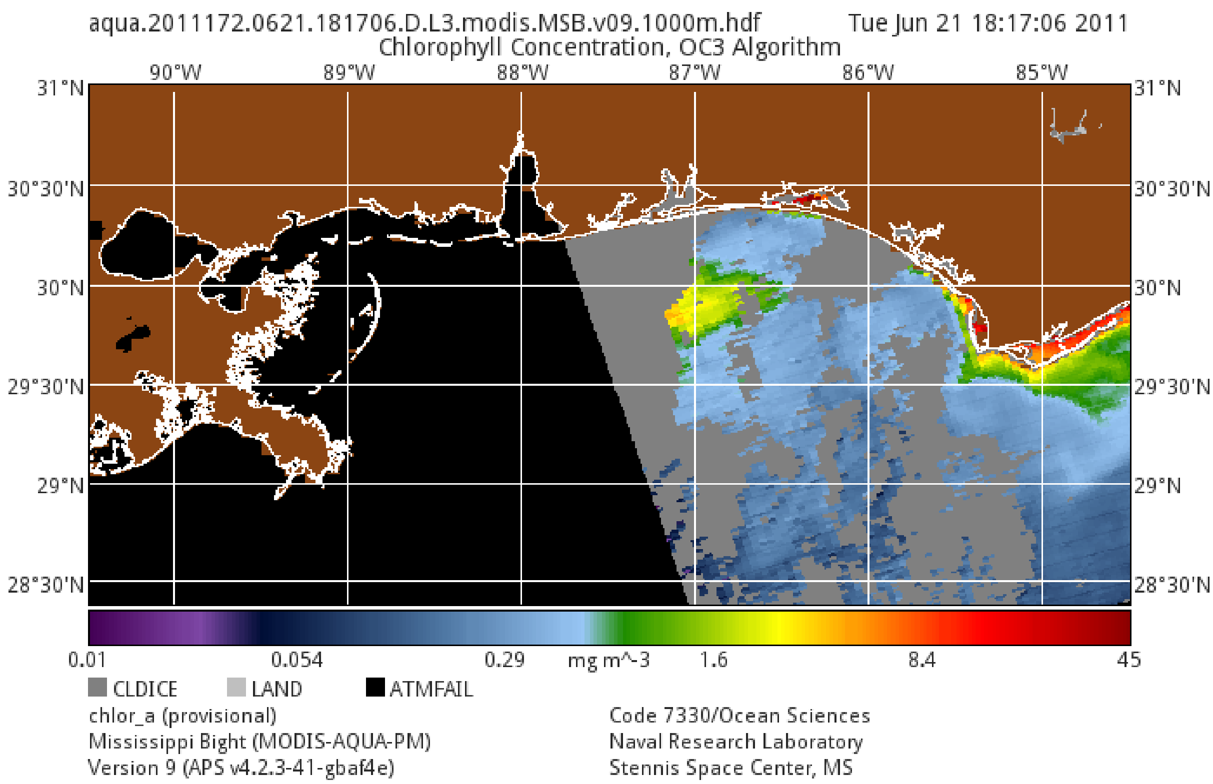
```
$ cat rolling.lst
aqua.2011170.0619.182921.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011170.0619.183008.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011171.0620.191209.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011172.0621.181706.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011172.0621.195511.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011173.0622.185951.D.L3.modis.MSB.v09.1000m.hdf
aqua.2011175.0624.184739.D.L3.modis.MSB.v09.1000m.hdf
$ ofile=aqua.2011170.0619.2011175.0624.D.L4_RL.modis.MSB.v09.1000m.hdf
$ rm -f $ofile
$ imgMean -F rolling.lst -T 6 -H chlor_a,Kd_488_lee,l2_flags -f 1,0,0 -o
$ofile
```

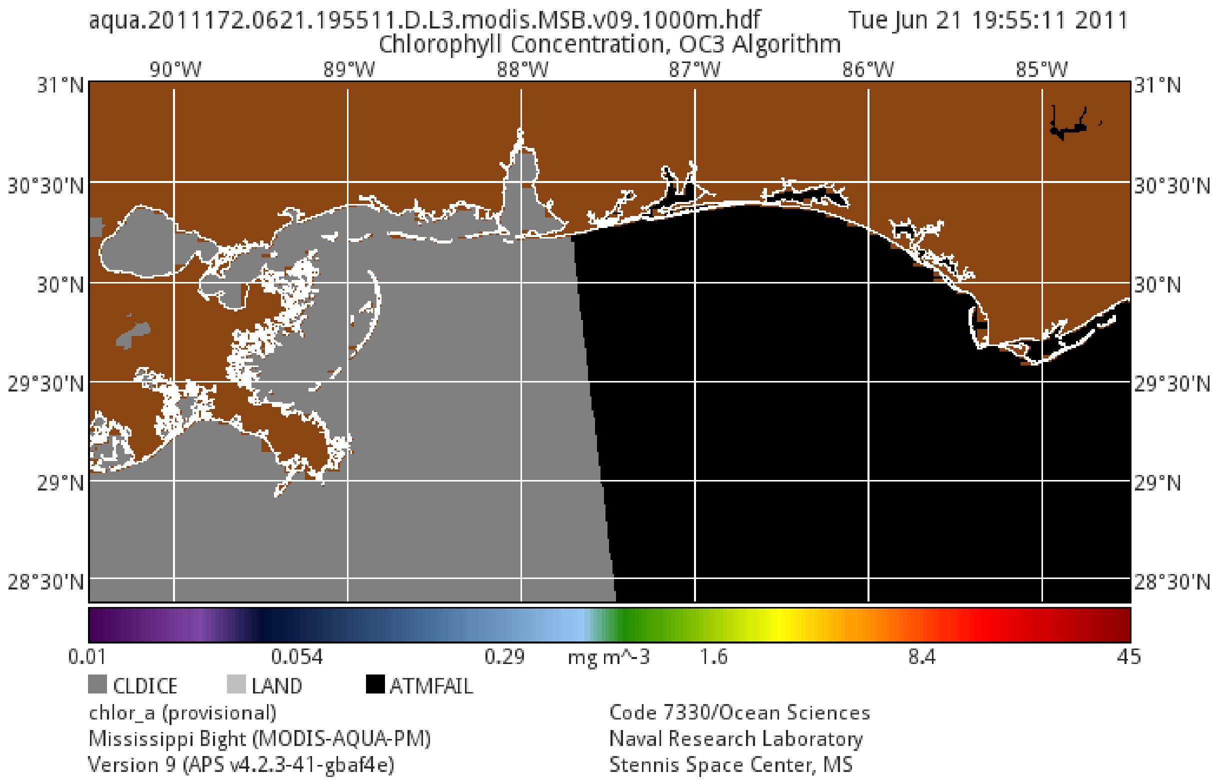
First we'll show you the six images that cover the first six days. Note, we are mising day 174 and have two passes for days 170 and 172.

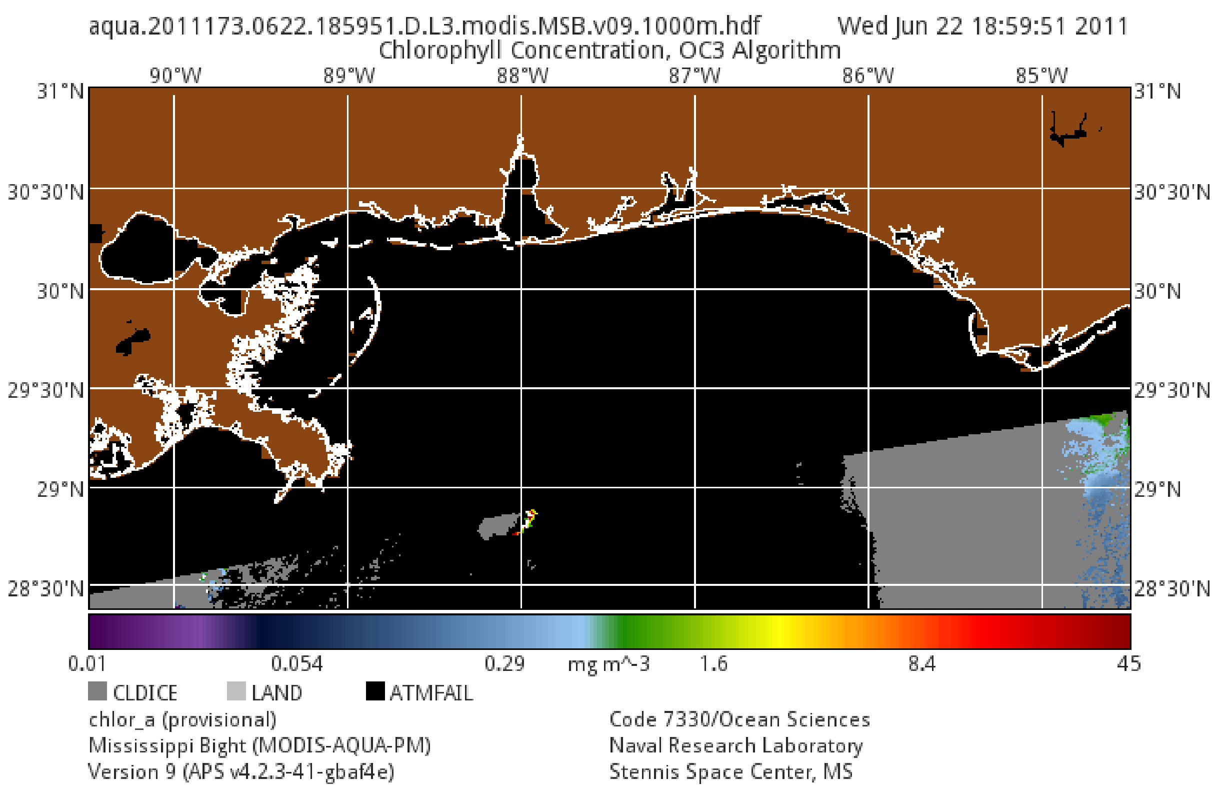


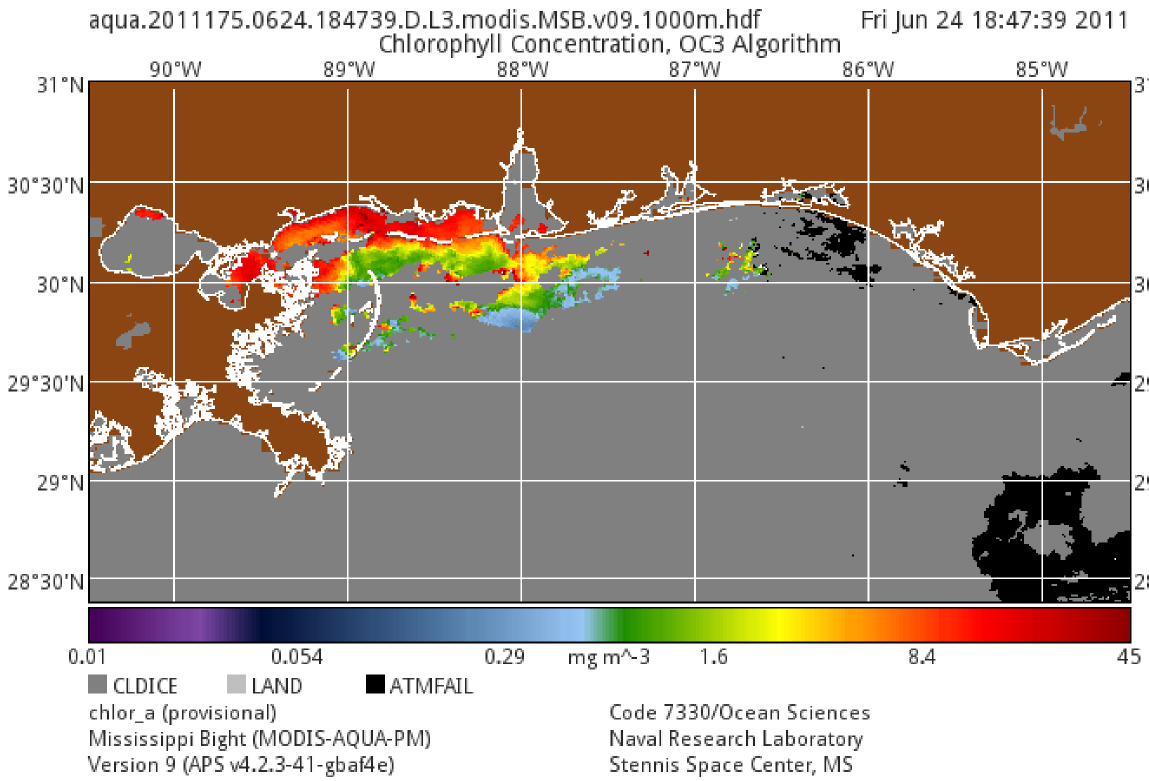



aqua.2011171.0620.191209.D.L3.modis.MSB.v09.1000m.hdf
Mon Jun 20 19:12:09 2011
Chlorophyll Concentration, OC3 Algorithm
90°W
89°W
88°W
87°W
86°W
85°W
31°N
30°30'N
30°N
29°30'N
29°N
28°30'N
0.01
0.054
0.29
mg m^-3
1.6
8.4
45
CLDICE
LAND
ATMFAIL
chlor_a (provisional)
Mississippi Bight (MODIS-AQUA-PM)
Version 9 (APS v4.2.3-41-gbaf4e)
Code 7330/Ocean Sciences
Naval Research Laboratory
Stennis Space Center, MS

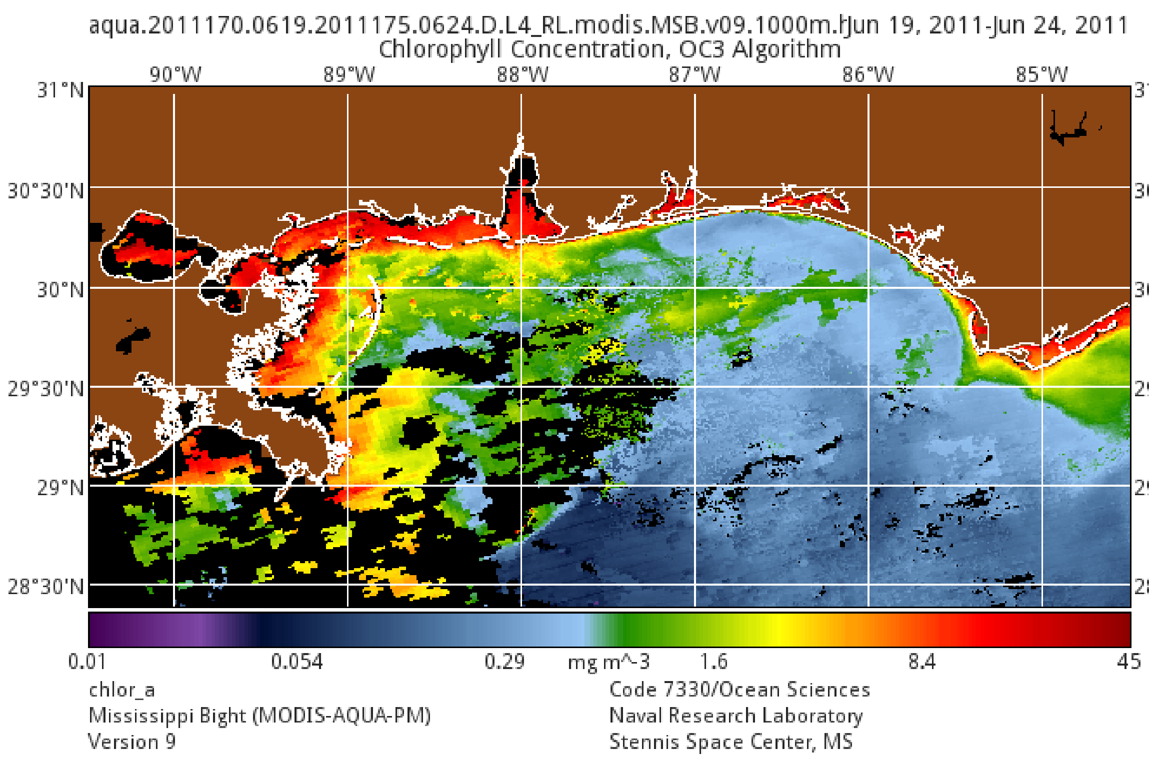

aqua.2011172.0621.181706.D.L3.modis.MSB.v09.1000m.hdf
Tue Jun 21 18:17:06 2011
Chlorophyll Concentration, OC3 Algorithm
90°W
89°W
88°W
87°W
86°W
85°W
31°N
30°30'N
30°N
29°30'N
29°N
28°30'N
0.01
0.054
0.29
mg m^-3
1.6
8.4
45
CLDICE
LAND
ATMFAIL
chlor_a (provisional)
Mississippi Bight (MODIS-AQUA-PM)
Version 9 (APS v4.2.3-41-gbaf4e)
Code 7330/Ocean Sciences
Naval Research Laboratory
Stennis Space Center, MS


aqua.2011172.0621.195511.D.L3.modis.MSB.v09.1000m.hdf
Tue Jun 21 19:55:11 2011
Chlorophyll Concentration, OC3 Algorithm
90°W
89°W
88°W
87°W
86°W
85°W
31°N
30°30'N
30°N
29°30'N
29°N
28°30'N
0.01
0.054
0.29
mg m^-3
1.6
8.4
45
CLDICE
LAND
ATMFAIL
chlor_a (provisional)
Mississippi Bight (MODIS-AQUA-PM)
Version 9 (APS v4.2.3-41-gbaf4e)
Code 7330/Ocean Sciences
Naval Research Laboratory
Stennis Space Center, MS


aqua.2011173.0622.185951.D.L3.modis.MSB.v09.1000m.hdf
Wed Jun 22 18:59:51 2011
Chlorophyll Concentration, OC3 Algorithm
90°W
89°W
88°W
87°W
86°W
85°W
31°N
30°30'N
30°N
29°30'N
29°N
28°30'N
0.01
0.054
0.29
mg m^-3
1.6
8.4
45
CLDICE
LAND
ATMFAIL
chlor_a (provisional)
Mississippi Bight (MODIS-AQUA-PM)
Version 9 (APS v4.2.3-41-gbaf4e)
Code 7330/Ocean Sciences
Naval Research Laboratory
Stennis Space Center, MS



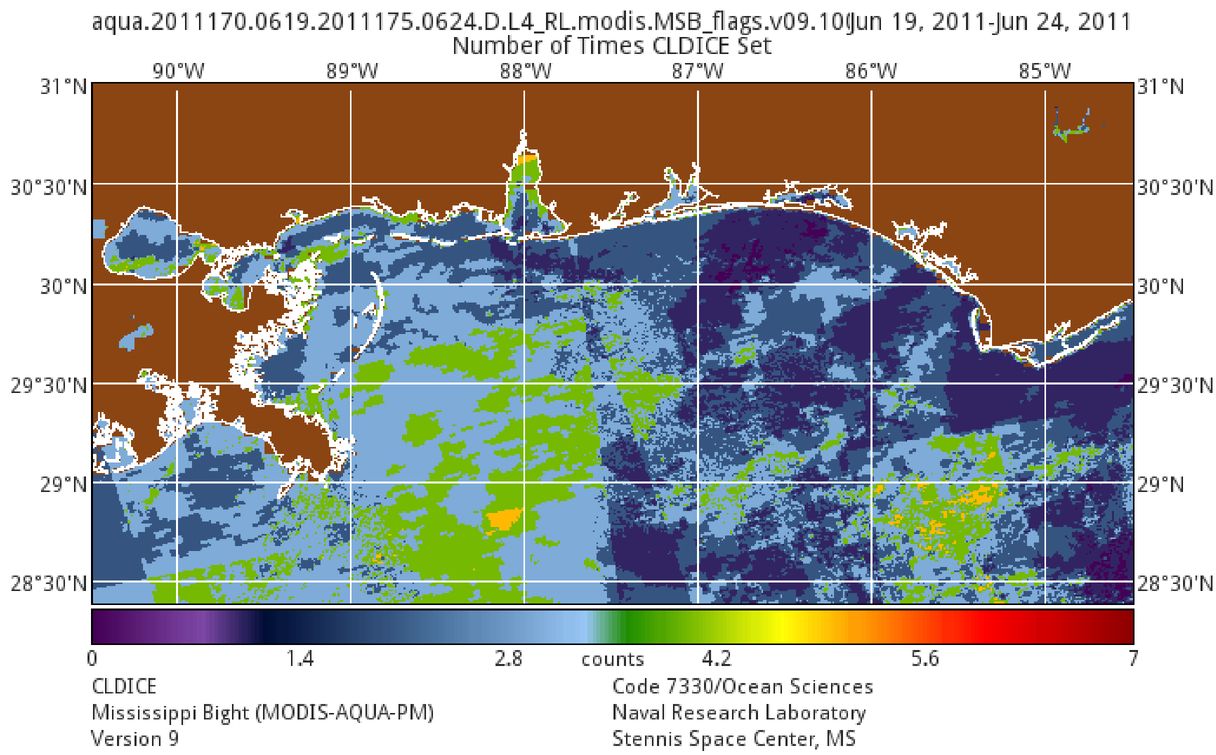
The result of our six-day “rolling” composite.

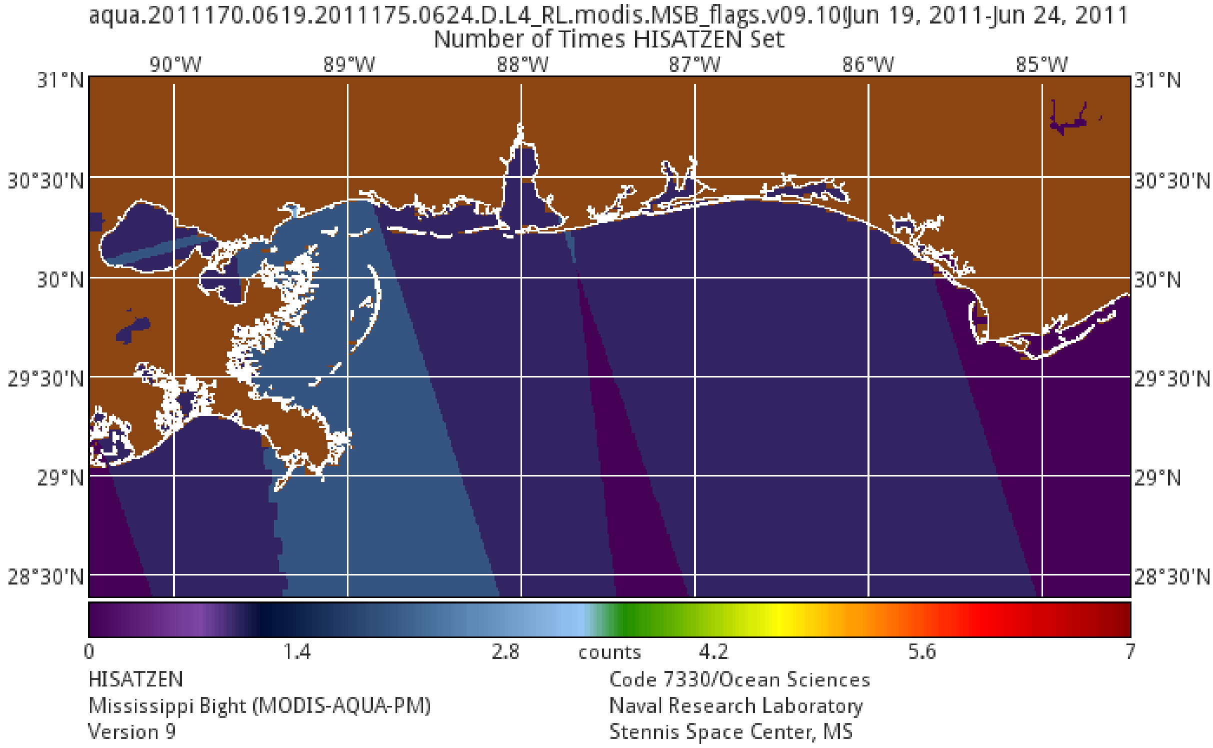


Producing a Flag Set Count

If we add the -k option, to the six-day composite from the section above **imgMean** will also include products which are the counts of the number of times a flag was set. For each input image which has the CLIDICE flag set, for example, there will be a product with the same name. This product will contain the number of input images for which the flag was set.

```
$ ofile=aqua.2011170.0619.2011175.0624.D.L4_RL.modis.MSB.v09.1000m.hdf
$ imgMean -F rolling.lst -T 6 -H chlor_a,Kd_488_lee,l2_flags -k -f 1,0,0 -o
$ofile
$ imgBrowse $ofile CLDICE $ofile.CLDICE.png
$ imgBrowse $ofile HISATZEN $ofile.HISATZEN.png
```





aqua.2011170.0619.2011175.0624.D.L4_RL.modis.MSB_flags.v09.10(Jun 19, 2011-Jun 24, 2011
Number of Times HISATZEN Set
90°W
89°W
88°W
87°W
86°W
85°W
31°N
30°30'N
30°N
29°30'N
29°N
28°30'N
0
1.4
2.8
counts
4.2
5.6
7
HISATZEN
Mississippi Bight (MODIS-AQUA-PM)
Version 9
Code 7330/Ocean Sciences
Naval Research Laboratory
Stennis Space Center, MS

Name

imgRead — dump information from images

Synopsis

imgRead [*options*] <*file*> <*product*> <*product*>...

Description

The program **imgRead** allows the user to retrieve data from an image at any desired position specified either by (latitude,longitude) pair or (line,sample) pair. The values are read as geophysical values and dumped to stdout. The user may select a single point or a square around the specified position.

Caveat

The latitude/longitude options can be used only with map projected files and must use the -m option. All others (Level-1, Level-2, etc.) can only use line,sample option, i.e. -x option.

Options

-b <size>	Use a box around the point of interest. Should be odd to place the point at the center (that is: 3, 5, 7, 9 ...)
-c	Output data in columns. If used with the -b option, this will output the average of the box in the column. Cannot be used with the -S option.
-f <%#.#f>	Used to control output formatting with the -b option. The number is the number of spaces for the entire number and the second number reprsents the number of decimal places. For example: -f %10.5f will output: xxxx.xxxxx Default format is %10.4f.
-F <flagName>	Show the flag specified as *flagName* as a 16 bit binary number. The output will be a string of 16 one's or zeros.
-g <outputFile>	Used to output data in a format acceptable by the GNU program graph. The argument outputFile should be the name of the output file to be created. The name of the product extracted will be appended to the outputFile name.
-h	This option is used to suppress the headers.
-m <mapFile:mapName>	This option is used when a mapped file will be read by the program. The first string should be that of the maps file followed by a colon and the name of the map. For example, -m default.maps:ChesepeakeBay.
-n <name>	Provides the name of the station which will be included in the header provided the *-s* option is used.
-p <x|y>	This option outputs a row (y) or a column (x) profile of the data.
-r <min,max>	Set the minimum and maximum range for the data used to calculate the output statistics when using the box option.
-s	Set to include the station name given by *-n* option in the output header.

-S	This will force the output to look similar to the Seadas output. Cannot be used with the -c option.
-t	Insert tabs between columns when using columnar output.
-x	Treat input values as samples and lines.
--help	Print out a small help guide.
--version	Print out version of software and quit.

Examples

To dump a series of points read in from file stations.dat and output to data.dat.

Example 30. Extracting Data From a Product File

```
$ more stations.dat
37.4502 -89.3403
37.5320 -89.3403
-99.0 -99.0
$ imgRead S1998100175129.N3_HNAV_MSB nLw_412 nLw_443 < stations.dat > data.dat
$ more data.dat
  Latitude   Longitude Pixel  Line     nLw_412     nLw_443
   37.4502    -89.3403   302   142     -0.2040      0.0300
   37.5320    -89.3403   303   141     -0.0010      0.2160
```

Name

imgSmooth — apply filter, perform statistics

Synopsis

`imgSmooth` [*options*] <*ifile*> <*ofile*> [<*product*> <*product*>...]

Description

This program is used to perform a transformation that gives each pixel in an image a new value that is a function of the pixels in its immediate neighborhood. The replacement function may be either the mean or the median.

In addition, the standard deviation of the pixels used to calculate the new pixel value may be calculated, along with the maximum and minimum pixel value. The function is applied to the input image by sliding a window over the image and applying the function to the pixels that fall underneath the window, in order to replace the center pixel in the window with the newly calculated value. If the window dimensions are even, then the center pixel is the pixel to the top and left of center. Invalid pixels and pixels flagged by a l2_flags mask are ignored during processing. The output file name may be the same as the input file name.

Options

-a
: Aggregate, slide window across image so that the pixels covered by the window used to create adjacent pixels does not overlap. This will create a smaller image.

  For example:

  - a 10 x 10 product with a 2 x 2 mask size will produce a 5 x 5 output product.
  - an 11 x 11 product will produce a 6 x 6 output product

  This option cannot be used with iterate, in addition the output file name must be different than the input file name since the product dimensions will change.

-d <num>
: Set maximum number of iterations. Note, iterations start after the first pass over the data.

-i <num>
: Iterate, continue smoothing until standard deviation falls below "num" threshold.

-j <num>
: Set l2_flags mask. Default: 523 = ATMFAIL,LAND,HIGLINT,CLDICE.

-k
: When iterating, compare average standard deviation of entire image to threshold. Default: compare greatest standard deviation of entire image.

-l
: Use l2_flags if available.

-m <num>
: Use *num* X *num* window.

-M
: Do median filter. Default: mean filter.

-n
: Create minimum neighbor image.

-q
: Use quick sort when performing median filter. Default: radix sort. Quick sort may be faster than radix sort for small mask sizes such as 2 x 2 or 3 x 3.

-S
: Create standard deviation of neighborhood image.

-v	Increase verbosity.
-x	Create maximum neighbor image.
--help	Print out a small help guide.
--version	Print out version of software and quit.

Examples

This smooths only the rrs_412 image

Name

imgStat — calculate statistics and data quality products of an image, and creates histograms.

Synopsis

`imgStat` [*options*] <*ifile*> [<*product*> <*product*>...]

Description

imgStat is used compute statistics and quality of data. It can generate histograms, a quality product - based on l2_flags, and quality attributes.

A histogram is generated for all products listed on the command line (or all available products in the file, if none given). The histogram information is written to the file as a series of attributes attached to each product (if the -a option is given) or dumped to the screen (if the -V option is given). In addition, an Encapsulated Postscript (.eps) graphic is created and stored in the input file (provided the -a option is given). The APS command **hdf** can be used to extract the plot.

Options

-a

Append histogram information to the *ifile*. Due to restrictions with the HDF format, if a file already has historgram information, it can not be updated.

-B isp=<isp>, iep=<iep>, isl=<isl>, iel=<iel>, irp=<irp>, irl=<irl>

Defines a subsection of the input image.

isp the starting sample number

iep the ending sample number

isl the starting line number

iel the ending line number

irp the replication factor along the samples dimension(not implemented)

irl the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat. If set to a negative number each sample is repeated the number of times equal to the absolute value of that number. Thus a positive irp is used to reduce or shrink the image and a negative irp is used to enlarge or magnify the image.

-B nlat=<nlat>, slat=<slat>, wlon=<wlon>, elon=<elon>, irp=<irp>, irl=<irl>

Defines a geographical subsection of input image.

nlat the latitude of most North Western point

slat the latitude of most South Eastern point

wlon the longitude of most North Western point

elon the longitude of most South Eastern point

irp	the replication factor along the samples dimension(not implemented)
irl	the replication factor along the lines dimension(not implemented)

The irp/irl indicates the number of samples/lines to skip or repeat (see previous -B description).

-f <function> Sets the scaling to one of the following:

lin	linear
log	logarithm on x-axis
loglog	logarithm on x and y-axes

-F <name> Name of input mask product , defaults to *l2_flags*

-n <nbins> Defines the number of bins to use; defaults to 100.

-r <min,max> Sets the defined range of data for the histogram. If not set, this will be obtained from the data file. First by looking of the attribute *validRange* and then to *browseRanges*.

-v Verbose output

-V Write out the histogram information to stdout. By default, the output is tab delimited.

--help Print out a small help guide.

--version Version information

Examples

This example will calculate statistics on the t_sen_443 product in the hdf file.

Example 31. Computing Statistics For a Product

```
$ imgStat -V -f lin -r 0.5,1.0 -n 25
 aqua.2011187.0706.110316.D.L3.modis.AAO.v09.1000m.hdf t_sen_443
Histogram for t_sen_443 (dimensionless)
mean 0.774723   median 0.78445
t_sen_443       # pixels       pdf     cum pdf means
  0.5100             0  0.0000    0.00    0.0000
  0.5300             0  0.0000    0.00    0.0000
  0.5500             0  0.0000    0.00    0.0000
  0.5700             0  0.0000    0.00    0.0000
  0.5900             0  0.0000    0.00    0.0000
  0.6100             0  0.0000    0.00    0.0000
  0.6300             0  0.0000    0.00    0.0000
  0.6500             0  0.0000    0.00    0.0000
  0.6700             0  0.0000    0.00    0.0000
  0.6900             0  0.0000    0.00    0.0000
  0.7100           385  0.0176    1.76    0.7042
```

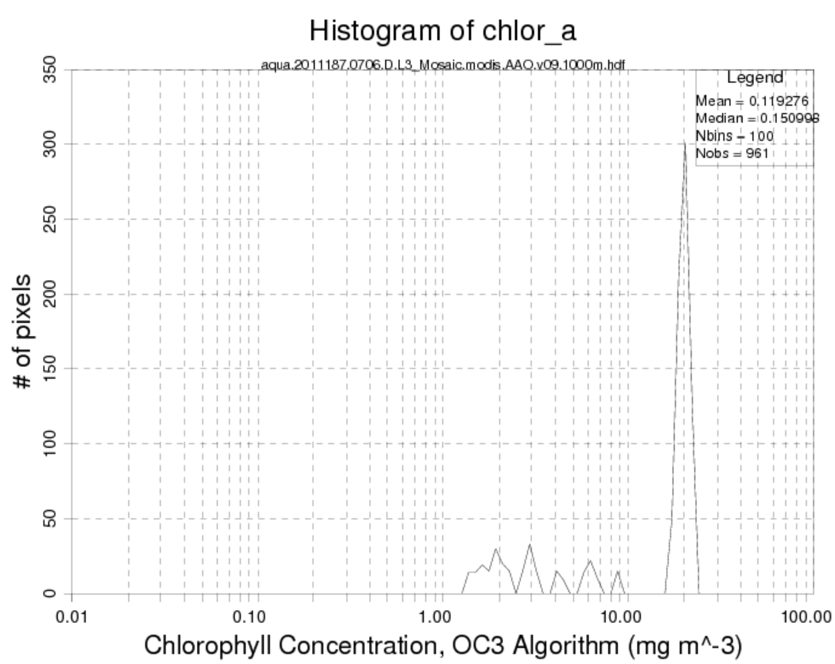
```
  0.7300          1627  0.0745    9.21    0.7180
  0.7500           389  0.0178   10.99    0.7325
  0.7700          3155  0.1445   25.44    0.7572
  0.7900         12767  0.5846   83.90    0.7849
  0.8100          3515  0.1610  100.00    0.7921
  0.8300             0  0.0000  100.00    0.0000
  0.8500             0  0.0000  100.00    0.0000
  0.8700             0  0.0000  100.00    0.0000
  0.8900             0  0.0000  100.00    0.0000
  0.9100             0  0.0000  100.00    0.0000
  0.9300             0  0.0000  100.00    0.0000
  0.9500             0  0.0000  100.00    0.0000
  0.9700             0  0.0000  100.00    0.0000
  0.9900             0  0.0000  100.00    0.0000
```

This example will calculate statistics on the chlor_a product in the hdf file and then extracts the plot from the file.

Example 32. Producing a histogram plot

```
$ imgStat -a aqua.2011187.0706.D.L3_Mosaic.modis.AAO.v09.1000m.hdf chlor_a
$ hdf aqua.2011187.0706.D.L3_Mosaic.modis.AAO.v09.1000m.hdf export
 chlor_a_hist.eps
exported 10289 bytes to chlor_a_hist.eps
```

Figure 17. Histogram of chlor_a produced by imgStat



Name

imgTSeries — compute difference between product(s) in two files.

Synopsis

`imgTSeries` [*options*] <*ifile*> <*roi*> <*ofile*> <*product*>

Description

imgTSeries is used to perform statistics on a region of interest (roi)and output the results in an ASCII file. The region of interest is defined by a geographical polygon (lat/lon). The polygonal region of interest is given a name and must indicate the number of points in the polygon. Several region of interests may be given in a single file.

Options

-d	Debug output
-h	Add a header to the CSV file prior to each product results
-l	Do not mask out pixels over land (by default land pixels are ignored)
-L <name>	Name of the input land mask file product. Defaults to `$APS_DATA/landmask.dat`
-M draw=<0|1>,name=<name>,masks=<"NAME1;NAME2;NAME3">	Controls how mask flags are used. draw — whether to use masks: 0-no, 1-yes (default is 1). name — name of mask product (default is 'l2_flags') masks — names of masks to overlay. Multiple mask names must be separated by a semi-colon ";".. Examples: **-M masks="CLDICE;ATMFAIL"**
-o <file.csv>	Name of file to receive CSV output
-r <min,max>	Set the minimum and maximum of the range of data to consider. Defaults to values set in "validRange" attribute.
-v	Verbose output
--help	Print out a small help guide.
--version	Print out version of software and quit.

Examples

Using a file that contains four geographical regions of the MissBight and a Level-3 MODIS image from the that regions, a CSV file will be generated to determine the statistical mean and standard deviation of each regional. The MissBight.blotch file used in this example is available in the share/aps directory of APS. It defines four regions.

Example 33. Regional statistics

```
$ file=aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf
$ imgTSeries -h -o MSB_rrs.csv $file MissBight.blotch \
         rrs_412 rrs_443 rrs_488 rrs_547
$ more MSB_rrs.csv
File,UNIX Epoch,Date/Time,Blotch,Product,Total Pixels,Number Used,Percent
 Coverage,Mean,Stdev,Sum,Sumsq
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,LakePonchatrain,rrs_412,1602,992,61.9226,0.0041,0.0012,4.0359,0.0179
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,MissBightShelf,rrs_412,27248,27247,99.9963,0.0013,0.0008,34.6813,0.0609
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,MissBightDeep,rrs_412,26520,26520,100.0000,0.0027,0.0008,71.5561,0.2117
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,Mobile Bay,rrs_412,917,674,73.5005,0.0048,0.0017,3.2020,0.0173
File,UNIX Epoch,Date/Time,Blotch,Product,Total Pixels,Number Used,Percent
 Coverage,Mean,Stdev,Sum,Sumsq
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,LakePonchatrain,rrs_443,1602,992,61.9226,0.0064,0.0014,6.3721,0.0428
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,MissBightShelf,rrs_443,27248,27247,99.9963,0.0020,0.0009,54.1551,0.1300
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,MissBightDeep,rrs_443,26520,26520,100.0000,0.0030,0.0006,78.9479,0.2451
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,Mobile Bay,rrs_443,917,673,73.3915,0.0069,0.0020,4.6413,0.0348
File,UNIX Epoch,Date/Time,Blotch,Product,Total Pixels,Number Used,Percent
 Coverage,Mean,Stdev,Sum,Sumsq
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,LakePonchatrain,rrs_488,1602,992,61.9226,0.0102,0.0018,10.0826,0.1057
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,MissBightShelf,rrs_488,27248,27247,99.9963,0.0027,0.0013,72.8304,0.2402
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,MissBightDeep,rrs_488,26520,26520,100.0000,0.0029,0.0004,77.9777,0.2343
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,Mobile Bay,rrs_488,917,673,73.3915,0.0097,0.0024,6.5270,0.0671
File,UNIX Epoch,Date/Time,Blotch,Product,Total Pixels,Number Used,Percent
 Coverage,Mean,Stdev,Sum,Sumsq
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,LakePonchatrain,rrs_547,1602,992,61.9226,0.0173,0.0027,17.1913,0.3051
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,MissBightShelf,rrs_547,27248,27247,99.9963,0.0025,0.0019,68.8701,0.2763
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,MissBightDeep,rrs_547,26520,26520,100.0000,0.0015,0.0003,38.7501,0.0590
aqua.2010063.0304.192110.D.L3.modis.MSB.v09.1000m.hdf,1267752070,03/04/10
 19:21:10,Mobile Bay,rrs_547,917,674,73.5005,0.0170,0.0031,11.4914,0.2026
```

Name

maps — interactive program to manipulate maps

Synopsis

`maps`

`maps` [*options*] <*mapfile*> <*map*>

Description

The program **maps** is an interactive program used to manipulate image maps. It is *not* intended to be very robust. It was originally designed to test the Aps_MapsXXX set of routines. But is also the only way to currently create and write an image map to a file for use in the areas scripts.

When the program is executed, the user is provided with a prompt to receive commands. The quick summary of the commands can be displayed using the "help" command. Below is a quick summary of the commands.

exit quit

Quits the program.

help

Prints a simple help menu. This list all the valid commands and one line on their purpose.

load

Adds any maps found in the user-specified file to the internal list.

save

Saves the internal list of maps to the user-specified file.

create

Creates a new map. This option will prompt the user for various parameters of the image map.

edit

Edit an image map to change the parameters.

copy

Copy a map to a new name.

zoom

Make a zoom copy of a map. The user selects the map to be copied, the destination name of the newly created map and the zoom factor.

delete

Deletes the user-specified map from the internal list.

dist_xy

Compute the distance in meters between two points on a given map. The map projection must be set prior to using this option.

dump

Dump the current navigation structure. A debugging option.

list

Lists all the maps in the internal list.

current

Lists the currently set image map.

setmap

Sets the user selected map to the current image map.

show

Prints out the parameters for a user-specified image map.

toll x y

Used to convert the user specified image coordinates (x,y) to geographic coordinates (lon,lat).

toxy lon lat

Used to convert the user specified geographic coordinates (lon,lat) to image coordinates (x,y).

toij lon lat

Used to convert the user specified geographic coordinates (lon,lat) to map coordinates (i,j).

onmap lon lat

Used to determine if the user specified geographic coordinates (lon,lat) on the image map.

boundary

Using the currently set image map, print a bounding polygon.

grid

Using the currently set image map, a world grid (lon,lat) is converted to image coordinates. The results are printed as a simple table.

dist_xy x1 y1 x2 y2

Computes the distance in meters between the two given image coordinates (x1,y1) and (x2,y2).

This program will be replaced by a more robust and GUI-oriented version some time in the future.

version

Prints out the version of this program.

Options

-b	Print out boundary of specified <map>.
-c	Print out the center longitude/latitude of a specified <map>.
-C <n>	Print out the cornet longitude/latitude of a specified <map>. Where *n* is 0 for upper left corner, 1 for upper right corner, 2 for lower right corner, 3 for lower left corner.
-e	Print extension <map>.
-h	Print out image height of specified <map>.
-l	Print list of all maps in <mapfile>.
-s	Print "show" command for <map>.
-w	Print out image width of specified <map>.
--help	Print out a small help page.
--version	Print out version of software and quit.

Map Projections

The following is a list of map projections supported by APS.

Table 15. Map Projections

Num	Name	Num	Name
0	Geographic	16	Sinusiodal
1	Universal Transverse Mercator (UTM)	17	Equirectangular
2	State Plane Coordinates	18	Miller Cylindrical
3	Albers Conical Equal Area	19	Van der Grinten
4	Lambert Conformal Conic	20	(Hotine) Oblique Mercator
5	Mercator	21	Robinson
6	Polar Stereographic	22	Space Oblique Mercator (SOM)
7	Polyconic	23	Alaska Conformal
8	Equidistant Conic	24	Interrupted Goode Homolosine
9	Transverse Mercator	25	Mollweide
10	Stereographic	26	Interrupted Mollweide
11	Lambert Azimuthal Equal Area	27	Hammer
12	Azimuthal Equidistant	28	Wagner IV
13	Gnomonic	29	Wagner VII

Num	Name	Num	Name
14	Orthographic	30	Oblated Equal Area
15	General Vertical Near-Side Perspective	99	User defined

Spheroids

The following is a list of spheriods supported by APS.

Table 16. Spheroids

Num	Name	Num	Name
0	Clarke 1866	16	Hough
1	Clarke 1880	17	Mercury 1960
2	Bessel	18	Modified Mercury 1968
3	International 1967	19	Sphere of Radius 6370997 meters
4	International 1909	20	Bessel 1841(Namibia)
5	WGS 72	21	Everest (Sabah Sarawak
6	Everest	22	Everest (India 1956)
7	WGS 66	23	Everest (Malaysia 1969)
8	GRS 1980	24	Everest (Malay Singapr 1948
9	Airy	25	Everest (Pakistan)
10	Modified Everest	26	Hayford
11	Modified Airy	27	Helmert 1906
12	WGS 84	28	Indonesian 1974
13	Southeast Asia	29	South American 1969
14	Australian National	30	WGS 60
15	Krassovsky		

Acknowledgements

The map projection software used by the APS came from the United States Geological Survey and is known as the General Cartographical Transformation Package. The APS uses version 2.0 of the C library.

Examples

For scripting purposes, the user might want to know the image dimensions of a an *image map*. To do that, we might use the following:

```
$ maps -w ~/aps_v6.4.4/etc/default.maps GulfOfMexico
2430
$ maps -h ~/aps_v6.4.4/etc/default.maps GulfOfMexico
1810
```

For the center and four corner points we have:

```
$ maps -c ~/aps_v6.4.4/etc/default.maps GulfOfMexico
-89.007407 25.061878
$ maps -C0 ~/aps_v6.4.4/etc/default.maps GulfOfMexico
-98.000000 31.000000
$ maps -C1 ~/aps_v6.4.4/etc/default.maps GulfOfMexico
-80.007407 31.000000
$ maps -C2 ~/aps_v6.4.4/etc/default.maps GulfOfMexico
-80.007407 18.811032
$ maps -C3 ~/aps_v6.4.4/etc/default.maps GulfOfMexico
-98.000000 18.811032
$ maps -C4 ~/aps_v6.4.4/etc/default.maps GulfOfMexico
invalid corner
```

An example of creating an equirectangular projection such that each pixel is exactly 0.01 degrees between each other. In other words, so that there are 100 pixels for each degree of latitude or longitude.

```
maps> create eq
Full Name? [ eq ] <return>
Code (3-letter)? [ eq ] <return>
Map Projection (? for list)? [ Mercator ] 17
Size of Map (w h)? 101 101
Pt1 of Map (lon lat)? -90 30
Pt1 of Map (x y)? [ 1 1 ] 51 51
Pt2 of Map (lon lat)? -91 31
Delta (pixels)? [ 100 ] <return>
Aspect ratio? [ 1.0 ] <return>
Datum (? for list)? [ WGS 84 ] <return>
Radius of Reference Sphere? [ 6378137.000000 ] <return>
Longitude of Central Meridian (lon)? 0
Latitude of True Scale (lat)? 0
False Easting? [ 0.0 ] <return>
False Northing? [ 0.0 ] <return>
maps> setmap eq
maps> show eq
   Name         eq
   Full Name    eq
   Code         eq
   Projection   17 (Equirectangular)
   Zone         62
   Datum        12 (WGS 84)
   Parameters:
    0:  6378137.000000 (Radius of Reference Sphere)
    1:  0.000000
    2:  0.000000
    3:  0.000000
    4:  0.000000 (Longitude of Central Meridian)
    5:  0.000000 (Latitude of True Scale)
    6:  0.000000 (False Eastings)
    7:  0.000000
    8:  0.000000
    9:  0.000000
```

```
    10:  0.000000
    11:  0.000000
    12:  0.000000
    13:  0.000000
    14:  0.000000
    Width        101
    Height       101
    Longitude_1  -90.0000
    Latitude_1    30.0000
    Pixel_1      51
    Line_1       51
    Longitude_2  -91.0000
    Latitude_2    31.0000
    Delta        100
    Aspect       1

Upper Left Corner       -90.5 30.5
Upper Middle Corner     -90 30.5
Upper Right Corner      -89.5 30.5
Middle Left             -90.5 30
Center                  -90 30
Middle Right            -89.5 30
Lower Left Corner       -90.5 29.5
Lower Middle Corner     -90 29.5
Lower Right Corner      -89.5 29.5

Map Distances
(   1,   1)     ( 101,   1)     95748.323212    957.483
(   1,  51)     ( 101,  51)     96236.863890    962.369
(   1, 101)     ( 101, 101)     96718.075759    967.181
(   1,   1)     (   1, 101)     111125.110164   1111.25
(  51,   1)     (  51, 101)     111125.110164   1111.25
( 101,   1)     ( 101, 101)     111125.110164   1111.25
maps> toll 1 1
-90.500000 30.500000
maps> toll 1 2
-90.500000 30.490000
maps> toll 1 3
-90.500000 30.480000
maps> toll 2 1
-90.490000 30.500000
maps> toll 3 1
-90.480000 30.500000
```

An example of making a “high-resolution” MODIS map of a previously created one kilometer map for the MissBight region.

```
maps> load $APS_ETC/default.maps
loaded 20 maps
maps> zoom
Name? MissBight
Name of new map? MissBight250
```

```
Zoom factor? 4
maps> show MissBight250
   Name         MissBight250
   Full Name    Mississippi Bight
   Code         MSB
   Projection    5 (Mercator)
   Zone         62
   Datum        12 (WGS 84)
   Parameters:
    0:  6378137.000000 (Semi-Major Axis)
    1:  6356752.314245 (Semi-Minor Axis)
    2:  0.000000
    3:  0.000000
    4:  -87030000.000000 (Longitude of Central Meridian)
    5:  29030000.000000 (Latitude of True Scale)
    6:  0.000000 (False Eastings)
    7:  0.000000 (False Northings)
    8:  0.000000
    9:  0.000000
   10:  0.000000
   11:  0.000000
   12:  0.000000
   13:  0.000000
   14:  0.000000
   Width        2400
   Height       1200
   Longitude_1  -90.5000
   Latitude_1    31.0000
   Pixel_1      1
   Line_1       1
   Longitude_2  -84.5000
   Latitude_2    28.0000
   Delta        2396
   Aspect       1
maps>quit
```

An example generating a series of boundary files from all the maps in the map file.

```
$ cat ./gmtMaps.sh
#! /bin/bash -x

APS_DIR=$HOME/aps_v6.4.4/
APS_BIN=$APS_DIR/bin
mapFile=$1

list=$($APS_BIN/maps -l $mapFile)
for m in $list
do
    $APS_BIN/maps -b $mapFile $m > $m.pts
done

$ ./gmtMaps.sh $HOME/aps_v6.4.4/etc/default.maps
```

```
$ ls *.pts
Adriatic.pts  MissBight.pts Texas.pts
Gomex.pts     SouthAus.pts
```

Using GMT all the boundary files can be put over an world map to show all the regions that are in default.maps file.

```
$ ./gmtBoundary.sh overall.eps MissBight.pts Gomex.pts Adriatic.pts
SouthAus.pts Texas.pts
$ convert -density 144 overall.eps overall.png
$ file overall.png
overall.png: PNG image data, 1726 x 1016, 8-bit grayscale, non-interlaced
$ display overall.png
```

See the Automated Optical Processing System User's Guide for another interactive example of this program's usage.

Name

n2gen — calibrate and atmospherically correct ocean color data

Synopsis

n2gen *par=file*
- *or* -

n2gen *ifile=ifile* *ofile1=ofile1*
- *or* -

n2gen *ifile=ifile* *geofile=geofile* *ofile1=ofile1* *ofile=ofile* *tgtfile=tgtfile* *aerfile=aerfile* *metafile=metafile* [*def_l2prod_file=def_l2prod_file*] [*l2prod1=l2prod1*] [*ofile* [*#*] *=ofile* [*#*]] [*l2prod* [*#*] *=l2prod* [*#*]] [*spixl=spixl*] [*epixl=epixl*] [*dpixl=dpixl*] [*sline=sline*] [*eline=eline*] [*dline=dline*] [*ctl_pt_incr=ctl_pt_incr*] [*proc_ocean=proc_ocean*] [*proc_land=proc_land*] [*proc_sst=proc_sst*] [*mode=mode*] [*aer_opt=aer_opt*] [*stumpf_opt=stumpf_opt*] [*stumpf_wave=stumpf_wave*] [*aer_wave_short=aer_wave_short*] [*aer_wave_long=aer_wave_long*] [*aer_swir_short=aer_swir_short*] [*aer_swir_long=aer_swir_long*] [*aer_rrs_short=aer_rrs_short*] [*aer_rrs_long=aer_rrs_long*] [*aermodmin=aermodmin*] [*aermodmax=aermodmax*] [*aermodrat=aermodrat*] [*aer_angstrom=aer_angstrom*] [*taua=taua*] [*aer_iter_max=aer_iter_max*] [*mumm_alpha=mumm_alpha*] [*mumm_gamma=mumm_gamma*] [*mumm_epsilon=mumm_epsilon*] [*absaer_opt=absaer_opt*] [*glint_opt=glint_opt*] [*outband_opt=outband_opt*] [*oxaband_opt=oxaband_opt*] [*filter_opt=filter_opt*] [*filter_file=filter_file*] [*brdf_opt=brdf_opt*] [*gas_opt=gas_opt*] [*land_opt=land_opt*] [*land_ndvi=land_ndvi*] [*land_Lt=land_Lt*] [*cloud_opt=cloud_opt*] [*sharpen=sharpen*] [*iop_opt=iop_opt*] [*pol_opt=pol_opt*] [*polfile=polfile*] [*xcalfile=xcalfile*] [*xcal_opt=xcal_opt*] [*xcal_wave=xcal_wave*] [*resolution=resolution*] [*giop_wave=giop_wave*] [*giop_fit_opt=giop_fit_opt*] [*giop_aph_opt=giop_aph_opt*] [*giop_adg_opt=giop_adg_opt*] [*giop_bbp_opt=giop_bbp_opt*] [*giop_maxiter=giop_maxiter*] [*giop_aph_s=giop_aph_s*] [*giop_aph_w=giop_aph_w*] [*giop_adg_s=giop_adg_s*] [*giop_adg_w=giop_adg_w*] [*giop_bbp_s=giop_bbp_s*] [*giop_bbp_w=giop_bbp_w*] [*giop_aphs=giop_aphs*] [*giop_aphw=giop_aphw*] [*gsm_opt=gsm_opt*] [*gsm_fit=gsm_fit*] [*gsm_adg_s=gsm_adg_s*] [*gsm_bbp_s=gsm_bbp_s*] [*gsm_aphw=gsm_aphw*] [*gsm_aphs=gsm_aphs*] [*ndvi_wav=ndvi_wav*] [*rgb_wave=rgb_wave*] [*qaa_version=qaa_version*] [*qaa_opt=qaa_opt*] [*qaa_adg_s=qaa_adg_s*] [*qaa_wave=qaa_wave*] [*ofmt=ofmt*] [*ofloat=ofloat*] [*chloc2_wave=chloc2_wave*] [*chloc2_coef=chloc2_coef*] [*chloc3_wave=chloc3_wave*] [*chloc3_coef=chloc3_coef*] [*chloc4_wave=chloc4_wave*] [*chloc4_coef=chloc4_coef*] [*ocm_destripe_mean_len=ocm_destripe_mean_len*] [*ocm_destripe_chunk_size=ocm_destripe_chunk_size*] [*ocm_destripe_cloud_threhold=ocm_destripe_cloud_threhold*] [*met1=met1*] [*met2=met2*] [*met3=met3*] [*ozone1=ozone1*] [*ozone2=ozone2*] [*ozone3=ozone3*] [*land=land*] [*water=water*] [*demfile=demfile*] [*icefile=icefile*] [*sstfile=sstfile*] [*no2file=no2file*] [*alphafile=alphafile*] [*tauafile=tauafile*] [*cldfile=cldfile*] [*calfile=calfile*] [*vcal_opt=vcal_opt*] [*offset=offset*] [*gain=gain*] [*albedo=albedo*] [*glint=glint*] [*absaer=absaer*] [*sunzen=sunzen*] [*satzen=satzen*] [*epsmin=epsmin*] [*epsmax=epsmax*] [*tauamax=tauamax*] [*nlwmin=nlwmin*] [*hipol=hipol*] [*wsmax=wsmax*] [*windspeed=windspeed*] [*windangle=windangle*] [*pressure=pressure*] [*ozone=ozone*] [*watervapor=watervapor*] [*maskland=maskland*] [*maskbath=maskbath*] [*maskcloud=maskcloud*] [*maskglint=maskglint*] [*masksunzen=masksunzen*] [*masksatzen=masksatzen*] [*maskhilt=maskhilt*] [*maskstlight=maskstlight*] [*sl_frac=sl_frac*] [*sl_pixl=sl_pixl*]

Description

This program is capable of performing atmospheric correction of top-of-atmosphere (TOA) radiances from several ocean remote sensing, spaceborne spectrometers, including MERIS, MODIS, OCM, SeaWiFS and deriving atmospheric and bio-optical properities using identical algorithms for all sensors. Data input format and sensor identification are automatically determined by the program. Sensor dependent details such as band-pass-weighted quantities are included in a sensor-specific external data file, and pre-computed sensor-specific look-up tables are provided for Rayleigh scattering and Rayleigh-aerosol transmittance. Aerosol model tables make use of sensor-specific coefficients, with some adjustment of the model epsilons to correct for deviations from sensor center wavelengths.

Products

This table contains all the products (don't get dizzy) which are available from this one program! Most outputs are 2-D arrays stored in an HDF file as a Scientific Data Sets (SDS) with the given name. The products which contain *nnn* are available at each wavelength of the given sensor. For MERIS these are: 413, 443, 490, 510, 560, 620, 665, 681, 709, 754, 762, 779, 865, 885, 900. For MODIS these are: 412, 443, 469, 488, 531, 547, 555, 645, 667, 678, 748, 859, 869, 1240, 1640, 2130. For OCM these are: 414, 441, 486, 511, 556, 669, 769, 865. For SeaWiFs these are: 412, 443, 490, 510, 555, 670, 765, and 865. Note that some products have multiple names for the same exact product. This is due to the nature of HDF's SDS interface wherein each SDS must have a unique name. Duplicate names are used for compatability between several systems which use the same code (SeaDAS, SeaWiFS Project, and NRL).

Product	Description
rrs_*nnn*	remote sensing reflectance at *nnn* nm
nLw_*nnn*	normalized water-leaving radiance at *nnn* nm
Lw_*nnn*	water-leaving radiance at *nnn* nm
Lr_*nnn*	Rayleigh radiance at *nnn* nm
L_q_*nnn*	polarization radiance at *nnn* nm, q-component
L_u_*nnn*	polarization radiance at *nnn* nm, u-component
polcor_*nnn*	polarization correction at *nnn* nm
La_*nnn*	aerosol radiance at *nnn* nm
TLg_*nnn*	TOA glint radiance at *nnn* nm
tLf_*nnn*	foam (white-cap) radiance at *nnn* nm
Lt_*nnn*	calibrated TOA radiance at *nnn* nm
Ltir_*nnn*	calibrated TOA radiance at *nnn* nm
rhot_*nnn*	TOA reflectance at *nnn* nm
brdf_*nnn*	BRDF coefficient at *nnn* nm
t_sol_*nnn*	Rayleigh-aerosol transmittance,sun to ground at *nnn* nm
t_sen_*nnn*	Rayleigh-aerosol transmittance,ground to sensor at *nnn* nm
t_oz_sol_*nnn*	ozone transmittance,sun to ground at *nnn* nm
t_oz_sen_*nnn*	ozone transmittance,ground to sensor at *nnn* nm
taua_*nnn*	aerosol optical depth at *nnn* nm
tau_*nnn*	same as taua_*nnn*

Product	**Description**
angstrom_*nnn*	aerosol angstrom coefficents (*nnn*,865) nm
eps_*nnn_lll*	ratio of *nnn* to *lll* single-scattering aerosol radiances
Es_*nnn*	extra-terestrial surface irradiance at *nnn* nm
rhos_*nnn*	surface reflectance at *nnn* nm
rhom_*nnn*	water + aeorsol reflectance at *nnn* nm (MUMM)
t_o2_*nnn*	total oxygen transmittance at *nnn* nm
t_h2o*nnn*	total water vaport transmittance at *nnn* nm
dpol_*nnn*	transmittance at *nnn* nm
alpha	polarization rotation angle
a_*nnn*	total absorption at *nnn* nm using default IOP algorithm (see iop_opt)
bb_*nnn*	backscatter at *nnn* nm using default IOP algorithm (see iop_opt)
a_*nnn*_carder	total absorption at *nnn* nm using Carder algorithm
aph_*nnn*_carder	phytoplankton absorption at *nnn* nm using Carder algorithm
adg_*nnn*_carder	detris/gelbstuff absorption at *nnn* nm using Carder algorithm
bb_*nnn*_carder	backscatter at *nnn* nm using Carder algorithm
b_*nnn*_carder	total scattering at *nnn* nm using Carder algorithm
c_*nnn*_carder	beam attenuation at *nnn* nm using Carder algorithm
c_length_*nnn*_carder	beam attenuation length at *nnn* nm using Carder algorithm
a_*nnn*_gsm	total absorption at *nnn* nm using GSM algorithm
aph_*nnn*_gsm	phytoplankton absorption at *nnn* nm using GSM algorithm
adg_*nnn*_gsm	detris/gelbstuff absorption at *nnn* nm using GSM algorithm
bb_*nnn*_gsm	backscatter at *nnn* nm using GSM algorithm
bbp_*nnn*_gsm	particulate backscatter at *nnn* nm using GSM algorithm
b_*nnn*_gsm	total scattering at *nnn* nm using GSM algorithm
c_*nnn*_gsm	beam attenuation at *nnn* nm using GSM algorithm
a_*nnn*_las	total absorption at *nnn* nm using Loisel and Stramski algorithm
b_*nnn*_las	total scattering at *nnn* nm using Loisel and Stramski algorithm
bb_*nnn*_las	backscatter at *nnn* nm using Loisel and Stramski algorithm
bbp_*nnn*_las	particulate backscatter at *nnn* nm using Loisel and Stramski algorithm

Product	Description
c_*nnn*_las	beam attenuation at *nnn* nm using Loisel and Stramski algorithm
a_*nnn*_niwa	total absorption at *nnn* nm using NIWA algorithm
bb_*nnn*_niwa	backscatter at *nnn* nm using NIWA algorithm
a_*nnn*_lmi	total absorption at *nnn* nm using LMI algorithm
aph_*nnn*_lmi	phytoplankton absorption at *nnn* nm using LMI algorithm
adg_*nnn*_lmi	detris/gelbstuff absorption at *nnn* nm using LMI algorithm
bb_*nnn*_lmi	backscatter at *nnn* nm using LMI algorithm
a_*nnn*_qaa	total absorption at *nnn* nm using QAA algorithm
aph_*nnn*_qaa	phytoplankton absorption at *nnn* nm using QAA algorithm
adg_*nnn*_qaa	detris/gelbstuff absorption at *nnn* nm using QAA algorithm
bb_*nnn*_qaa	backscatter at *nnn* nm using QAA algorithm
bbp_*nnn*_qaa	particulate backscatter at *nnn* nm using QAA algorithm
b_*nnn*_qaa	total scattering at *nnn* nm using QAA algorithm
c_*nnn*_qaa	beam attenuation at *nnn* nm using QAA algorithm
flag_qaa	quality flags from QAA algorithm
mod_rrs_qaa	modeled rrs at 640 nm from QAA algorithm
a_*nnn*_pml	total absorption at *nnn* nm using Plymouth Marine Laboratory algorithm
aph_*nnn*_pml	phytoplankton absorption at *nnn* nm using Plymouth Marine Laboratory algorithm
adg_*nnn*_pml	detris/gelbstuff absorption at *nnn* nm using Plymouth Marine Laboratory algorithm
bb_*nnn*_pml	backscatter at *nnn* nm using Plymouth Marine Laboratory algorithm
bbp_*nnn*_qaa	particulate backscatter at *nnn* nm using Plymouth Marine Laboratory algorithm
c_645	beam attenuation at 645 nm using Gould algorithm
cp_645	particulate attenuation at 645 nm using Gould algorithm
wmass	water mass classification using Gould algorithm
water_mass	water mass classification image using Gould algorithm
PIM_gould	particulate inorganic matter using Gould algorithm
POM_gould	particulate organic matter using Gould algorithm
TSS_gould	total suspened particles using Gould algorithm

Product	Description
aph_412_gould	phytoplankton absorption at 412 nm using Gould algorithm
aph_443_gould	phytoplankton absorption at 443 nm using Gould algorithm
asd_412_gould	sediment and detrital absorption at 412 nm using Gould algorithm
asd_443_gould	sediment and detrital absorption at 443 nm using Gould algorithm
ad_412_gould	detrital absorption at 412 nm using Gould algorithm
ad_443_gould	detrital absorption at 443 nm using Gould algorithm
ag_412_gould	gelbstuff absorption at 412 nm using Gould algorithm
ag_443_gould	gelbstuff absorption at 443 nm using Gould algorithm
ap_412_gould	particulate absorption at 412 nm using Gould algorithm
ap_443_gould	particulate absorption at 443 nm using Gould algorithm
as_412_gould	sediment absorption at 412 nm using Gould algorithm
as_443_gould	sediment absorption at 443 nm using Gould algorithm
class_ward_owmc	OWMC Ward Classification
class_k_owmc	OWMC K_means Classification
class_34k_w_owmc	OWMC merged Wards and K_means (34K+W) Classification
aph_443_stumpf	phytoplankton absorption at 443 nm using Stumpf's algorithm
adg_412_stumpf	DOM and gelbstuff absorption at 412 nm using Stumpf's algorithm
adg_555_stumpf	DOM and gelbstuff absorption at 555 nm using Stumpf's algorithm
a_412_stumpf	total absorption at 412 nm using Stumpf's algorithm
a_555_stumpf	total absorption at 555 nm using Stumpf's algorithm
chl_oc2	chlorophyll-a concentration using OC2 algorithm
chl_oc3	chlorophyll-a concentration using OC3 algorithm
chl_oc4	chlorophyll-a concentration using OC4 algorithm
chlor_a	chlorophyll-a concentration using sensor-specific default
chl_stumpf	chlorophyll-a concentration using Stumpf's algorithm
chl_carder	chlorophyll-a concentration using Carder's algorithm
Kd_532	diffuse attenuation at 532 nm using 490/555 ratio

Product	Description
K_length_532	diffuse attenuation at 532 nm using 443/555 ratio
Kd_*nnn*_lee	diffuse attenuation at *nnn* nm using Lee algorithm
Kd_490_morel	diffuse attenuation at 490 nm using Morel Eq8
Kd_490_morel_ok2	diffuse attenuation at 490 nm using Morel OK2
Kd_490_mueller	diffuse attenuation at 490 nm using Mueller
Kd_490_obpg	diffuse attenuation at 490 nm using OBPG
Kd_PAR_morel	diffuse attenuation (PAR) using Morel algorithm (1st optical depth)
Kd_PAR_lee	diffuse attenuation (PAR) using Lee algorithm (1st optical depth)
Zsd_lee	Secchi depth, Lee algorithm
Zsd_morel	Secchi depth, Morel algorithm
Zeu_lee	Euphotic depth, Lee algorithm
Zeu_morel	Euphotic depth, Morel algorithm
par	photosynthetically active radiation
ipar	instantaneous photosynthetically active radiation
arp	instantaneous absorbed radiation by photoplankton
flh	fluourescene line height
flh_ruhul	fluourescene line height, Ruhul algorithm
ebi_ruhul	extreme bloom index, Ruhul algorithm
rbd_ruhul	red band difference, Ruhul algorithm
depth	water depth index
aerindex	aerosol index
aer_model_min	minimum bounding aerosol model #
aer_model_max	maximum bounding aerosol model #
aer_model_ratio	model mixing ratio
aer_num_iter	number of aerosol iterations, NIR correction
glint_coeff	glint radiance normalized by solar irradiance
l2_flags	level-2 processing flags
epsilon	retreived epsilon used for model selection
eps_78	same as epsilon
cloud_albedo	cloud albedo at 865 nm
rho_cirrus	cirrus reflectance (1380 nm)
tindx_morel	turbidity index, Morel
tindx_morel	turbidity index, Shi and Wang
latitudes	latitudes (-90.0 to 90.0)
longitudes	longitudes (-180.0 to 180.0)
solz	solar zenith angle

Product	Description
sola	solar azimuth angle
senz	satellite zenith angle
sena	satellite azimuth angle
pixnum	pixel number
detnum	detector number
mside	mirror side
windspeed	magnitude of wind at 10 meters
zwind	zonal wind speed at 10 meters
mwind	meridional wind speed at 10 meters
windangle	wind direction at 10 meters
water_vapor	precipital water concentration
humidity	relative humidity
pressure	barometric pressure
ozone	ozone concentration
no2_tropo	tropospheric NO2
no2_strat	stratospheric NO2
no2_frac	Fraction of tropospheric NO2 above 200 m
ice_frac	Ice Fraction, 0=no ice, 1=all ice
fsol	solar distance correction (1-D, not an image)
ndvi	normalized difference vegetation index
evi	enhanced vegetation index
smoke	smoke index
height	terrain height
BT_39	Brightness Temperature at 3.9 um
BT_40	Brightness Temperature at 4.0 um
BT_11	Brightness Temperature at 11 um
BT_12	Brightness Temperature at 12 um
sst	Sea Surface Temperature
sst4	4um Sea Surface Temperature
sst_osisaf	Sea Surface Temperature, OSISAF algorithm
sst_mc_navo	Sea Surface Temperature, NAVO MCSST algorithm
sst_nl_navo	Sea Surface Temperature, NAVO NLSST algorithm
sst_nl2_navo	Sea Surface Temperature, NAVO NLSST extended algorithm
bias_sst	Sea Surface Temperature Bias
bias_sst4	4um Sea Surface Temperature Bias
stdv_sst	Sea Surface Temperature Standard Deviation
stdv_sst4	4um Sea Surface Temperature Standard Deviation

Product	Description
salinity	salinity using Ladner's algorithm
salinity_wright	salinity using Wright's algorithm
horiz_vis	horizontal diver visibility
horiz_vis_645	horizontal diver visibility using 4.8/c_645
vert_vis	vertical diver visibility
lidar	LIDAR penetration from Kd

Options

Each of these options listed below must be placed on the command line or in the parameter files as keyword=value pairs. If the enviornment variables (see Environmental Variables below) are defined, then the only required keyword=value pairs are ifile and ofile1. The others have reasonable defaults.

par Input parameter file to be used in the specific command mode **n2gen par="parfile"**. The parameter file is a text file containing the user-defined keyword=value pairs, each on a single line.

Input/Output File

n2gen has the ability to output up to four seperate files each with their own list of products (ofile1, ofile2, ofile3, and ofile4 contain the names of each file, and l2prod1, l2prod2, l2prod3, and l2prod4 contain the list of desired products for each output file). This allows the user to seperate products into various output files.

ifile Directory path and filename of the input Level-1 data product.

ilist A text file containing a list of files to process.

geofile An ancillary file that contains the geolocation information for the input file.

hkmfile For MODIS processing defines the input 500m data file

okmfile For MODIS processing defines the input 1KM data file

ofile1 Directory path and filename of the output Level-2 HDF file containing the products specified in the defaults file or by the **l2prod1** keyword. If not defined, then the output file will be automatically generated based on information about the file. Default file name will be: <satellite>.YYYYJJJ.MMDD.HHMMSS.(D|N|B|U).L2.<sensor>.<area>.v<version>.<resolution>.<extension>

area Defines the “area” string of the output file. Only used if ofile is not defined.

extension Defines the “extension” of the output file and format. Must be either 'hdf', 'nc', or 'h5'. Only used if ofile is not defined.

l2prod1 The product names (see main description section above) to be output to ofile1.

suite Defines a particular suite of products to produce.

pversion processing version string.

ofmt *n* This keyword defines the type of file output. If set to 0 or 1, a flat binary type of format is written. The default value of 2, creates the standard NASA HDF file format. A value of 3 will produce an NRL APS file format.

ofloat *n* For NRL/APS formatted files this option will cause all products to be written in floating point.

Input File Parameters

n2gen has the ability to subsection/subsample the input file and to control the number of control points in each output file. If not specified, n2gen will work on the entire input file.

The user may provide the sample/line locations from the file (spixl,epixl,sline,eline) or provide a geographical box (north,south,east,west) or a point and a box size (south,west,xbox,ybox).

spixl Starting pixel to be processed (default=1)

epixl End pixel to be processed (default=0, meaning the last pixel in the scan line)

dpixl Pixel subsampling interval (default=1)

sline Starting line number to be processed (default=1)

eline Ending line number to be processed (default=0, meaning the last line in file)

dline Line subsampling interval (default=1)

north Define Northern most latitude of geographical box

south Define Southern most latitude of geographical box

east Define Eastern most longitude of geographical box

west Define Western most longitude of geographical box

xbox Cross-track distance (in pixels) from center location (given by south and west)

ybox Along-track distance (in lines) from center location (given by south and west)

ctl_pt_incr Control-point pixel increment for lon/lat arrays (0=optimize, default=1)

Processing Options

These parameters control the overall processing.

proc_ocean Set to 1 for ocean processing; 0 for none. Set to 2 to process all pixels as if ocean.

proc_land Set to 1 for land processing; 0 for none. That is, produce the surface reflectance (rhos) product and any other user selected land products like NDVI or EVI. Since true_color is based on rhos, this option should be set to 1.

proc_sst Set to 1 for SST processing; 0 for none.

atmocor Set to 1 for full atmospheric correction processing; 0 for none.

Calibration Control Options

These keywords control the calibration of the input data.

calfile Directory path and filename of the input calibration table file. The default is $CAL_HDF_PATH.

vcal_opt Control for calibration modification: 0 - use gain, offset from the calibration table 1 - use the gain (as multipliers of the existing gains in the table file) from the input (gain) and offset from the calibration table 2 - use offsets from the input (offset) and the gains from the calibration table 3 - use both gain (as multipliers of the existing gains in the table file) and offset from the input (gain & offset)

gain Calibration gain factors to multiply the gain values read from the $APS_DATA/*sensor*/*sensor*_table.dat file. Defaults to [0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0].

offset Calibration gain offset factors to substitute for values read from the calibration table. Defaults to [1.0,1.0,1.0,1.0,1.0,1.0,1.0,1.0].

Ancillary Input Files

These keywords control the ancilliary input files. If the environment variables have been set (see Environmental Variables), then these keywords have reasonable defaults. If they are not set, then the user *must* define these: land, water, met1, and ozone1.

land Directory path and filename of the land-mask input file (when land_opt=0). Default is $APS_DATA/common/landmask.dat

water Directory path and filename of the shallow water mask input file used for setting the l2_flags bit to indicate shallow water (defined as 30m) areas (McClain et al., 1995). Default is $APS_DATA/watermask.dat

met1 Directory path and filename of the climatological product or the near-real-time (NRT) meteorological ancillary data product available for the nearest time preceding the time of ifile product's first scan line. If met1 is the climatological file, then met2 and met3 will not be used, otherwise, see met2 for logic. Default is $MSL1_DATA/CLIMATOLOGY.MET.

met2 Directory path and filename of the NRT meteorological ancillary data product available for the nearest time following the time of ifile product's first scan line's. If met2 is not specified (null) and met1 is a NRT product, then met2 will be set to met1. If met1 <> met2 and the scan line's date and time, fall between the times of met1 and met2, get_ancillary will use met1 and met2 to generate the interpolated meteorological values (if the scan line's date and time fall before those of met1, an error occurs). If met1 = met2 and the scan line's date and time fall before met2, get_ancillary will use only met2 to generate the meteorological values. If met2 <> met3 and the scan line's date and time fall between the times of met2 and met3, get_ancillary will use met2 and met3 to generate the interpolated meteorological values (if the scan line's date and time fall after those of met3, an error occurs). If met2 = met3 and the scan line's date and time fall after met2, get_ancillary will use only met2 to generate the meteorological values.

met3 Directory path and filename of the NRT meteorological ancillary data product for the nearest time following the time of ifile product's last scan line. If met3 is not specified (null) and met1 is a NRT product, then met3 will be set to met2 and the logic specified in met2 will be applied.

ozone1 Directory path and filename of the climatological product or the NRT ozone ancillary data product available for the nearest time preceding the time of ifile product's first scan line. If ozone1 is the climatological file, then ozone2 and ozone3 will not be used, otherwise see ozone2 for logic. (For TOVS data, the center point time is used to represent the time of that product.) Defaults to $APS_DATA/CLIMATOLOTY.OZONE.

ozone2 Directory path and filename of the NRT ozone ancillary data product available for the nearest time following the time of ifile product's first scan line's. If ozone2 is not specified (null) and ozone1 is a NRT product, then ozone2 will be set to ozone1. If ozone1 <> ozone2 and the scan line's date and time, fall between the times of ozone1 and ozone2, get_ancillary will use ozone1 and ozone2 to generate the interpolated ozone values (if the scan line's date and time fall before those of ozone1, an error occurs). If ozone1 = ozone2 and the scan line's date and time fall before ozone2, get_ancillary will use only ozone2 to generate the ozone values. If ozone2 <> ozone3 and the scan line's date and time fall between the times of ozone2 and ozone3, get_ancillary will use ozone2 and ozone3 to generate the interpolated ozone values (if the scan line's date and time fall after those of ozone3, an error occurs). If ozone2 = ozone3 and the scan line's date and time fall after ozone2, get_ancillary will use only ozone2 to generate the ozone values. (For TOVS data, the center point time is used to represent the time of that product.)

ozone3 — Directory path and filename of the NRT ozone ancillary data product for the nearest time following the time of ifile product's last scan line. If ozone3 is not specified (null) and ozone1 is a NRT product, then ozone3 will be set to ozone2 and the logic specified in ozone2 will be applied. (For TOVS data, the center point time is used to represent the time of that product.)

cldfile — This parameter should point to a MODIS Cloud Mask file which will be used to perform cloud masking.

demfile — digital elevation map file.

sstfile — SST reference file.

no2file — no2 ancillary file.

alphafile — alpha510 climatology file.

Algorithm control options

These keywords modify and/or select the algorithms used to correct the input data or change certain thresholds used for various tests.

ocm_destripe_mean_len — The number of pixels used in a running mean which gives smooth data to derive correction coefficients from. Default = 30 pixels. OCM only.

ocm_destripe_chunk_size — The number of lines used for each chunk of the image used to derive striping correction coefficients. Each chunk of lines is averaged into a mean scan for that chunk. A running mean of this mean scan gives the estimate of de-striped data from which correction coefficients are derived. All scan lines are used from the processed segment of the image and the resulting correction coefficients are averaged together (excluding noise) to produce the coefficients used to apply to the entire image. Default=100. OCM only.

ocm_destripe_cloud_threshold — The Lt(865) cloud/land threshold for OCM destriping. This threshold is used to exclude pixels from the destripe algorithm used to derive correction coefficients for all bands. Defaults to 0.75. OCM only.

filter_opt — Option for filtering the L1B data with the method specified in filter_file. 1=apply filtering , 0=do not apply filtering. (Default for OCTS=1, Default for others=0).

filter_file — Directory path and filename of the filter file that contains the filter method and information to be applied to the L1B data when *filter_opt* is set to 1. (Default=$APS_DATA/*sensor*/*sensor*_filter.dat).

outband_opt	SeaWiFs out-of-band corrections. 0 = no correction, 1 = no Lw correction, 2 = full correction. (Default = 1 for SeaWiFs, 0 for all others)
oxaband_opt	SeaWiFs/OCTS 764nm band Oxygen correction. 0 = off, 1 = on. (Default = 1 for SeaWiFs and OCTS, 0 for others)
glint_opt	Correct for residual glint radiance. 0 = off, 1 = on. (Default = 0)
land_opt	Masking for land. 0 = use landmask file (land=) and original Goddard interface, 1 = an NDVI-based test, 2 = use landmask file (land=) and APS LandMask interface, (Default = 0). See ocean color documentation for more information.
cloud_opt	Masking for clouds. 0 = albedo, 1 = reflectance ratio test. (Default = 0)
aer_opt	Option for aerosol calculation mode. The default is -3.

Value	Description
positive	Multi-scattering with fixed model
0	White aerosol extrapolation
-1	Multi-scattering with 2-band model selection
-2	Multi-scattering with 2-band RH-based model selection and iterative NIR correction
-3	Multi-scattering with 2-band model selection with iterative NIR correction
-4	Multi-scattering with fixed model pair (requires aermodmin, aermodmax, aermodrat specification)
-5	Multi-scattering with fixed model pair and iterative NIR correction (requires aermodmin, aermodmax, aermodrat specification)
-6	Multi-scattering with fixed angstrom (requires aer_angstrom specification)
-7	Multi-scattering with fixed angstrom and iterative NIR correction (requires aer_angstrom specification)
-8	Multi-scattering with fixed aerosol optical thickness (requires taua specification)
-9	Multi-scattering with 2-band model selection using Wang and Shi turbidity index (1.30) to switch between SWIR and NIR. (Requires SWIR bands

		and requires aer_swir_short, aer_swir_long, aer_wave_short, aer_wave_long)
	-10	Multi-scattering with MUMM correction and MUMM NIR calculation
	-11	Spectral optimization via Kuchinke (SeaWiFS-only)
	-12	Spectral matching via Gordon (SeaWiFS-only)
	-43	Multi-scattering with th 2-band model selection using SWIR bands to estimate NIR contribution
aer_wave_short	shortest sensor wavelength for aerosol model selection (default: 765 or similar)	
aer_wave_long	longest sensor wavelength for aerosol model selection (default: 865 or similar)	
aer_swir_short	shortest sensor wavelength for SWIR-based NIR Lw correction (default: -1)	
aer_swir_long	longest sensor wavelength for SWIR-based NIR Lw correction (default: -1)	
stumpf_opt	Correct Rrs using Stumpf 412 iteration. 0 = off, 1 = on. (Default = 0)	
stumpf_wave	List of wavelengths needed for Stumpf 412 iteration.	
aermodmin	lower-bounding model to use for fixed model pair aerosol option (default: -1)	
aermodmax	upper-bounding model to use for fixed model pair aerosol option (default: -1)	
aermodrat	ratio to use for fixed model pair aerosol option (default: 0.0)	
aer_angstrom	aerosol angstrom exponent for model selection	
aer_iter_max	The maximum number of aerosol iterations (Default = 10 for any NIR algorithm).	
mumm_alpha	water-leaving reflectance ratio for MUMM turbid water atmospheric correction.	
mumm_gamma	two-way Rayleigh-aerosol transmittance ratio for MUMM turbid water atmospheric correction.	
mumm_epsilon	aerosol reflectance ratio for MUMM turbid water atmospheric correction.	
absaer_opt	absorbing aerosol flagging option 0: use rhow constant 1: apply chlorophyll climatology to calculate rhow 2: 1+validate against nLw_412 climatology.	

brdf_opt	Bidirectional reflectance correction 0: no correction 1: Fresnel reflection/refraction correction for sensor path 3: Fresnel reflection/refraction correction for sensor + solar path 7: Morel f/Q + Fresnel solar + Fresnel sensor 15: Gordon DT + Morel f/Q + Fresnel solar + Fresnel sensor 19: Morel Q + Fresnel solar + Fresnel sensor.
gas_opt	gaseous transmittance bitmask selector 0: no correction 1: Ozone 2: CO2 4: NO2 8: H2O
iop_opt	desired inherent optical properity (IOP) model for use in downstream products (for example, Kd(lee), water_mass, visibility, etc.). The a, a_xxx, bb and bb_xxx products will contain the selected model outputs (a "default" IOP algorithm). It is contolled by these values:

0	None (products requiring a or bb will fail)
1	Carder
2	GSM
3	QAA
4	PML
5	NIWA
6	LAS
7	GIOP
8	LMI

polfile	polarization sensitivites filename leader.
pol_opt	polarization correction (sensor-specific) 0: no correction 1: only Rayleigh component is polarized 2: all radiance polarized like Rayleigh 3: only Rayleigh and Glint are polarized (MODIS default) 4: all radiance polarized like Rayleigh + Glint
rad_opt	radiation correction option (sensor-specific) 0: no correction 1: apply MERIS Smile correction
xcalfile	cross-calibration file.
xcal_opt	cross-calibration option (sensor-specific) comma separated list of option values, 1 per band, with bands listed in xcal_wave. 3: apply cross-calibration corrections (polarization and rvs) 2: apply cross-calibration polarization corrections 1: apply cross-calibration rvs corrections 0: no correction.
xcal_wave	wavelengths at which to apply cross-calibration. Comma separated list of sensor wavelength values associated with xcal_opt.
resolution	processing resolution (MODIS only) -1: standard ocean 1km processing 1000: 1km resolution including aggregated 250 and 500m land bands 500: 500m resolution including aggregated 250 land bands and replication for lower resolution bands 250: 250m resolution with replication for lower resolution bands
giop_adg_opt	Generic IOP model adg function type 0: tabulated (supplied via giop_adg_file) 1: exponential with exponent supplied via giop_adg_s) 2: exponential with exponent derived via QAA method 3: exponential with exponent derived via OBPG method

giop_adg_file	Generic IOP model, tabulated adg spectra (default=$APS_DATA/common/adg_default.txt).
giop_adg_s	Generic IOP model, spectral parameter for adg.
giop_aph_opt	Generic IOP model aph function type 0: tabulated (supplied via giop_aph_file) 2: Bricaud (read from $APS_DATA/common/aph_bricaud_1995.txt) 3: Ciotti (read from $APS_DATA/common/aph_ciotti_2002.txt w/size fraction supplied via giop_aph_s)
giop_aph_file	Generic IOP model, tabulated aph spectra (default=$APS_DATA/common/aph_default.txt).
giop_aph_s	eneric IOP model, spectral parameter for aph
giop_bbp_opt	Generic IOP model bbp function type 0: tabulated (supplied via giop_bbp_file) 1: power-law with exponent supplied via giop_bbp_s) 2: power-law with exponent derived via HAL method 3: power-law with exponent derived via QAA method 4: power-law with exponent derived via PML method 5: power-law with exponent derived via CIOTTI method 6: power-law with exponent derived via MM01 method 7: power-law with exponent derived via LAS method 8: Loisel and Stramski basis vector 9: Loisel and Stramski fixed vector 10: QAA fixed vector
giop_bbp_file	Generic IOP model, tabulated bbp spectra (default=$APS_DATA/common/bbp_default.txt).
giop_bbp_s	Generic IOP model, spectral parameter for bbp.
giop_maxiter	Generic IOP model iteration limit
giop_rrs_opt	Generic IOP model Rrs to bb/(a+bb) method 0: Gordon quadratic (specified with giop_grd) 1: Morel f/Q
giop_fit_opt	Generic IOP model optimization method 0: Amoeba optimization 1: Levenberg-Marquardt optimization 2: LU Decomposition (number of giop_wave elements must equal number parameters to fit) 3: SVD matrix inversion
giop_grd	Generic IOP model, Gordon Rrs to bb/(a+bb) quadratic coefficients (default=[0.0949,0.0794]).
giop_wave	sensor wavelengths for fitting Generic IOP model comma-seperated list, defaults is all visible bands (400-700nm)
tau_a	Aerosol optical thickness at 865 nm for fixed model. If tau_a > 0 and aer_opt > 0, then the input tau_a will be used to derive aerosol reflectance.
qaa_version	Select QAA algorithm version. Version 4 is based on Rrs@640 nm and Version 5 is based on Rrs@670.
qaa_opt	Source of 640 reflectance for use in QAA algorithm version 4. 0 is using the Rrs@640 (useful only for the Hi-Res MOIS and MERIS sensors). 1 is for using a synthetically derived Rrs@640.
qaa_wave	Used to define the sensor wavelengths that best are to be used for the QAA algorithm.

qaa_adg_s	Define the S parameter to use in the QAA algorithm. Default is 0.015.
lmi_wave	Used to define the sensor wavelengths that best are to be used for the LMI algorithm.
lmi_adg_s	Define the S parameter to use in the LMI algorithm. Default is 0.018.
lmi_aph_g	Define the g parameter to use in the LMI algorithm. Default is 70.
lmi_bbt	Define the bb parameters to use in the LMI algorithm. Default is [0.8,1.5].
sunzen	Solar zenith angle in degrees; threshold for setting the l2_flags bit to indicate large solar zenith angles (McClain et al., 1995). Default is 65.0.
satzen	Spacecraft zenith angle in degrees; threshold for setting the l2_flags bit to indicate large satellite zenith angles (McClain et al., 1995). Default is 56.0.
albedo	Cloud albedo for band 8 in percent; threshold for setting the l2_flags bit to indicate clouds or ice (McClain et al., 1995).
absaer	Absorbing aerosol threshold on aerosol index. Default is 0.5.
glint_thresh	Sun glint threshold (fraction of F0(865)); used in calculations for setting the l2_flags bit to indicate sun glint (McClain et al., 1995).
tauamax	Maximum 865 aerosol optical depth used for setting the l2_flags bit #5. (Default=0.3)
epsmin	Minimum epsilon to trigger atmospheric correction failure flag (default is 0.65).
epsmax	Maximum epsilon to trigger atmospheric correction failure flag (default is 1.35).
wsmax	Windspeed limit on white-cap correction. (Default=12.0 m/s).
sl_frac	Lt 865 threshold for stray-light correction. (SeaWiFs only.) (Default = 0.25)
sl_pixl	Number of LAC pixels over which stray-light correction is applied. 0 = no correction, -1 = program defaults (8 for GAC, 3 for LAC)

Masking keywords

These keywords select flags to be used as masks. A mask is a special flag that will cease execution on the pixel which passes the flag. For example, maskland will skip all land pixels.

maskland	Mask out land pixels: 0=off, 1=on. (Default=1).
maskbath	Mask out shallow water pixels: 0=off , 1=on. (Default=0).
maskcloud	Mask out cloud or ice pixels: 0=off, 1=on. (Default=1).
maskglint	Mask out sun glint pixels: 0=off, 1=on. (Default=1).
masksunzen	Mask out large solar zenith angle pixels: 0=off, 1=on. (Default=0).
masksatzen	Mask out large sensor zenith angle pixels: 0=off, 1=on. (Default=0).

maskhilt — Mask out pixels for which total radiance was greater than knee value: 0=off, 1=on. (Default=1).

maskstlight — Mask out stray light contaminated pixels: 0=off, 1=on. (Default=1).

Debugging controls

station_input — The name of an input station file. The format is UNIX text file with three columns. The first two are the sample and line location of the desired pixel and the last is an ASCII station name field. During each iteration of the atmosopheric correction information will be dumped to an output file designated by station_output

station_output — The name of an output station file. This will receive the data dumps for each station for each iteration of the atmospheric correction algorithm.

Environmental Variables

APS_DATA — Root directory for atmospheric correction data files. If this environmental variable is not set, then several ancilliary input file keywords will have to be defined. See ANCILLIARY INPUT FILES above.

Examples

This the minimum command line execution (assumes the enviornmental variables have been set as shown):

```
>
$ export APS_DATA=/home/aps/aps_v6.4.4/data
$ n2gen ifile=S2000148173615.L1A_HNAV ofile1=S2000148173615.L2_HNAV
```

In this example, the user has selected to output the remote sensing reflectance data into one file, the bio-optical products K_490 and chl_oc4 in another, and some of the Arnone IOP products in a third. It also turns off the masking of high Lt values and increases the cloud and ice threshold to 1.5.

```
>
$ n2gen ifile=S2000148173615.L1A_HNAV ofile1=S2000148173615.L2.rrs \
     l2prod1="rrs_412 rrs_443 rrs_490 rrs_510 rrs_555 rrs_670" \
     ofile2=S2000148173615.L2.bio l2prod2="chl_oc4 K_490" \
     ofile3=S2000148173615.L2.iop l2prod3="a_443_qaa bb_555_qaa" \
     albedo=0.3 maskhilt=0
```

Name

vc-aeronet — processes L1A data files for vicarious calibration information

Synopsis

`vc-aeronet.rb` *`input`*

Description

The **vc-aeronet.rb** takes the input data and performs a calibration. To begin, the **vc-aeronet.rb** examines the input Level-1 file to extract from it the date and time, the sensor name, the L1A calibration (look-up-table) number, and its set of wavelengths. Additionally, if the geo-location data is found in a separate file, that information is obtained.

Next, **vc-aeronet.rb** will repeat the following procedure on each site defined in the configuration file (default is `$APS_ETC/vc-aeronet.plist`). Based on the date obtained from the satellite, a query is made to obtain all the AERONET or MOBY data for that site for that given day. If no AERONET or MOBY data is available, the next site is examined.

From all the AERONET or MOBY, the closest entry (given a time window) for that day is identified. If no data for that day is identified, the next site is examined. For the record chosen, the AERONET or MOBY data is obtained for the wavelengths that match the satellite data. By default, the AERONET data requires a spectral shift to match. The MOBY data as obtained from NOAA has already been corrected for the satellite spectral response. The **rrs_model** program uses the AERONET nLw to obtain a hyperspectral remote sensing reflectance using a technique from Lee. This hyperspectral R_{rs} is then convolved with the sensor response of the input file.

Using this *in-situ* data (AERONET or MOBY), the **n2gen** program is run in “vicarious calibration” mode. In this mode, the atmospheric correction is applied to the input Level-1 data to obtain all the atmospheric components and normalization parameters that would produce a normalized water-leaving radiance, nL_w. At this point, the *in-situ* replaces the computed nL_w and all the normalization parameters and atmospheric components are added back to produce top-of-the-atmosphere radiance, L_t. To distinquish this top-of-the-atmosphere radiance from that of the satellite Level-1 inputs, the letter **v** is prefixed: vL_t. If there are any errors running **n2gen** processed is stopped for this site and the next is examined.

The file output from the **n2gen** can be of any size around the point for which the *in-situ* data resides, but only that point is used to the last step in the **vc-aeronet.rb** processing. That is the generation of plots, XML data files, comma-separated value files, and importation of the results into SAVANT.

Options

-a/--archive <directory>	move files created to <directory>
--aeronet	use aeronet.gsfc.nasa.gov as source of AERONET-OC data
-B/--csvsql	add CSV data to the SAVANT vical_lines table
-c/--config <file>	use <file> as the configuration file
-C/--csvfile <file>	use <file> as the CSV (results) file
--cache	cache AERONET-OC data from SQL data base to $HOME/AERONET
--database <cal>	use the SQL CalVal data base as source of AERONET-OC data

-d/--datfile [<file>]	produce an XML data (.plist) file, using <file> if provided
-D/--debug	produce debugging information.
-e/--epsfile [<file>]	produce an EPS plot file, using <file> if provided
-E/--error <directory>	use <directory> as the location to place files when an error occurs
-l/--list <file>	use <file> as a list of files to process
-L/--logfile <file>	use <file> as the log file
-R/--region <region>	use only <region> from configuration file
--verbose	Turn on messages.
--version	Print software version.
--help	Print help message.

Gain/Offset Options

These gain and offsets are applied when the original L_t data when the Level-1 file is pushed to the surface and are not to be confused with the resulting "gains" or ratios of vL_t/L_t that this program computes and writes to the CSV file.

-G/--gain <name>	Use <name> as part of the recipe when applying a specific gain/offset.
-g/--gains <num>	Use <num> as the gains when performing the vc matchup.
-o/--offset <num>	Use <num> as the offsets when performing the vc matchup.

Files

Though **vc-aeronet.rb** may create several files, the primary file of intest is the CSV file. This file places all the required information about each vicarious calibration matchup into a single line.

Table 17. Vicarious Calibration CSV File Format

Field	Description
name	name of in-situ data site
instrument	instrument number of in-situ data site
level	level of in-situ data
insitu time	date and time of insitu data
latitude	latitude of in-situ site (in decimal degress, -90 to 90)
longitude	longitude of in-situe site (in decimal degress, -180 to 180)
satfile	satellite file name
recipe	vicarious calibration recipe triplet
sattime	date and time of satellite data
processedVersion	processed version of satellite data
latitude	latitude of satellite point (in decimal degress, -90 to 90)

Field	Description
longitude	longitude of satellite point (in decimal degress, -180 to 180)
solz	solar zenith angle
sola	solar azimuth angle
senz	sensor zenith angle
sena	sensor azimuth angle
l2_flags	Level-2 processing flags
gain(#)	ration of vLt/Lt
Lt(#)	satellite top-of-the-atmosphere radiance
vLt(#)	top-of-the-atmosphere radiance (based on vnLw)
nLw(#)	satellite normalized water-leaving radiance
Lw(#)	satellite water-leaving radiance
normalization flag	2 if normalized, 1 if not
vnLw(#)	AERONET or MOBY input radiance
Lr(#)	Rayleigh radiance
La(#)	aerosol radiance
TLg(#)	glint radiance
tLf(#)	foam radiance
brdf(#)	BRDF
polcor(#)	polarization correction
t_sol(#)	transmittance, to sun
t_sen(#)	transmittance, to sensor
tg_sol(#)	transmittance, to sun
tg_sen(#)	transmittance, to sensor
t_o2(#)	Ozone transmittance
aer_model_min	minimum model number
aer_model_max	maximum model number
recnum	the record number
pixnum	the pixel number
detnum	the detector number
mside	the mirror side

SAVANT

When the -B/--csvsql option is used, the data computed from the **vc-aeronet.rb** data will be imported into SAVANT. A line of CSV data will be inserted into the **vical_lines** tables. See Files above for the layout. Additionally data will be written to the **vical_id** and **vical_dataz** tables.

Configuration File

The default configuration file for the **vc-aeronet.rb** is `$APS_ETC/vc-aeronet.plist`. When the -c/--config option is used, the user may select a different configuration file.

The format of this XML file is known as a properly list.

Table 18. Vicarious Calibration Configuration File Format

Name	Option	Description
logfile	-l/--logfile	file to receive logging information
logsize		maximum size of file before rolling
loglevel		level of logging: ERROR, WARN, INFO or DEBUG
archive	-a/--archive	path to use for archiving files
epsfile	-e/--epsfile	path of EPS file (blank value means auto-generated name)
datfile	-d/--datfile	path of data file (blank value means auto-generated name, plist format)
csvsql	-B/--csvsql	do we add the vical data to SAVANT (vical_lines, vical_id, vical_dataz)
recipes		array of selected recipes to run: std
boxsize		size of box around the in-situ point
sites		an array of information about each site
Levels		which AERONET data levels to use
ProdList		an array of products to generate

Recipes

The recipes parameter determine which processing options to run for a matchup. The value **std** means the standard atmospheric correction (Gordon/Wang, NIR iteration, no Stumpf iteration, bands 7 and 8).

A recipe triple consists of three parts: the first part is the radiometric calibration that is applied to the Level-1 data; the second part reflects any gains that may be applied to the data; and the last part is the atmospheric correction. The first part is dependent upon the input data and is determined as such. The second part has a single default option called **g00**. This setting is for the use of unity gains and zero offsets. That is no vicarious calibration is performed. The final part is related to the atmospheric correction and any adjustments to the *in-situ* data. Currently it can only have the values of as indicated above.

Since **vc-aeronet.rb** can only process a GOCI, MERIS, MODIS/AQUA, SeaWiFS and VIIRS/NPP Level-1 input, the following table will fully enumerate the possible triplets.

Table 19. Vicarious Calibration Recipes

Triplet	Description
l00-g00-std	GOCI Standard KORDI L1B calibration, unity gains, and standard atmospheric correction
l03-g00-std	MERIS 3rd-reprocessing L1B calibration, unity gains, and standard atmospheric correction
l06-g00-std	MODIS Collection 6 L1B calibration, unity gains, and standard atmospheric correction
l01-g00-std	SeaWiFS Standard NASA L1B calibration, unity gains, and standard atmospheric correction
l00-g00-std	VIIRS Standard NOAA L1B calibration, unity gains, and standard atmospheric correction

Archiving

When the -a/--archive option is used, the user may use special characters to define the auto-generated archive directory. These include: %R, %Y, %y, %J, %M, and %D.

Table 20. Vicarious Calibration --archive Special Characters

Name	Option	Description
Year	%Y	Replace with 4-digit start year of file.
Year	%y	Replace with 2-digit start year of file.
Day of Year	%J	Replace with 3-digit day of year.
Month	%M	Replace with 2-digit month.
Day	%D	Replace with 2-digit day of month.
Region	%R	Replace with region as defined in the configuration file.

For example, the user defines the archive as: `/home/aps/aeronet/%R/%Y/%J`. And processes a VIIRS Level-1A file from 20 January 2013 with start time of 18:16:15. Additionally, the XML data file and EPS graphics files are auto-generated, then one would find the following files. This pass covers the AERONET-OC WaveCIS site.

```
  $ ls -1 /home/aps/aeronet/WaveCIS/2013/020
  npp.2013020.0120.181615.D.L2.viirs.WaveCIS_Site_CSI_6.l00-g00-std.750m.hdf
  npp.2013020.0120.181615.D.L2.viirs.WaveCIS_Site_CSI_6.l00-g00-
std.750m.hdf.l00-g00-std.eps
  npp.2013020.0120.181615.D.L2.viirs.WaveCIS_Site_CSI_6.l00-g00-
std.750m.hdf.l00-g00-std.plist
```

Appendix A. Release Notes

The following notes are summaries of the changes to the APS software for each release. The most recent changes are at the top.

Release v6.4.4

This version was released on 29 August 2016.

- fix **imgMean** to handle `sst` units.
- **filefmt** recognizes directory arguments.
- default **imgMean** output to netCDF. That is, `-o` option is not used.
- update **data/modisa/n2gen_defaults.par** to fix issue with visibility products (at wrong wavelength).
- modify **imgBrowse**, **imgMap**, **imgMean** to handle OBPG files better.
- Fix **imgBrowse** to handle missing `Fahrenheit` calibrations when `otherUnits` indicates they exist.
- **apsCalValDB.rb** program has new options `--after`, `--before`, and `--between` which can be used to limit the AERONET-OC data which will inserted into database.
- **apsCalValDB.rb** program's (`--moby`) MOBY processing was updated to handle those files with a corrected ES set of columns.
- the vicarious calibration scripts were updated to make data insertion faster.
- updates to database tables.
  - a new **calval_products** table has been added.
  - the **calval_main** table was updated to replace **name** column with a foreign key to the **calval_products** table.
  - the **calval_main** table was updated to replace the foreign key **pdate_id** column with UNIX epoch time stamp. This allows the **calval_data** table to contain entry only related to the data.

Release v6.4.3

This version was released on 25 July 2016.

- a missing prototype for bb to b function was missing in get_lmi.c causing c_vvv_lmi to produce wrong results.
- Updates to database tables.
  - the **aeronet_data** table now contains a field for UNIX epoch. The table column order was also modified.
  - the **aeronet_data** md5 hash is now computed using the Postgres md5 function rather than through ruby.
- Several core libraries have been updated.
  - docbook-xsl-ns v1.79.1 replaces v1.79.0
  - fop v2.1 replaces v2.0

Release v6.4.2

This version was released on 19 July 2016.

- Fix APS Level-2 navigation control point. Corrects imgRead and imgBrowse, etc.
- Fix -Q option for PNG output for proper flipping
- New visibility products for LMI at 532nm (`vis_lmi_a`,`vis_lmi_bb`, `vis_lmi_c`,`vis_lmi_Kd`).
- Initial support for groups in APS IO library
- **imgBrowse**, **imgMap**, and **imgRead** can handle Goddard's new netCDF Level-2 files
- Add MODIS/Terra R2014.0 data support
- Several core libraries have been updated.
  - HDF5 v1.8.17 replaces HDF5 v1.8.12
  - HDF v4.2.12 replaces HDF v4.2.10
  - netCDF v4.4.1 replaces netCDF v4.3.1.1
  - PNG v1.6.23 replaces PNG v1.6.9

Release v6.4.1

This version was released on 31 March 2016.

- New option `-o format=` in **imgMap** and **imgMean** for setting output format.
- Update 250m MODIS data processing.
- Update 375m VIIRS data processing.
- The vicarious calibration for GOCI data is based on a cross-calibration with VIIRS data from the GOCI_CalSite_1 from 2014. This calibration was performed in April of 2016.

Table A.1. GOCI Vicarious Calibration

	412	443	490	555	660	680	745	865
v6.4	0.9726	0.9520	0.9258	0.8974	0.9007	0.8719	0.9430	1.0000

Release v6.4.0

This version was released on 29 February 2016.

- Updated **n2gen** to Goddard v8.9.10 (from v8.1.4). This corresponds to SeaDAS 7.3.
  - new time functions in libtimeutils
  - scale Rayleigh by the altitude of the sensor, assuming height of atmosphere=100km `scaleRayleigh = 1.0 - exp(-l1rec->alt/10); // Assume 10km is e-folding height`
  - new `fqfile` option for Morel F/Q BRDF parameters

  - new `chl_abi` ABI Chlorophyll (Shanmugam, 2011)
  - now use reflectance scale/offset for OLI counts to Lt on input
  - new `add_noise_sigma` option to add random normal noise to gain
  - new `band_shift_opt` option to apply bio-optical bandshift
  - new Harmful Algal Bloom products from Rick Stumpf.
  - new Ocean Primary Productivity products.
  - new CDOM products.
- **imgRGB** program has been retired.
- **true_color** product has been removed. Use rhos_xxx and imgBrowse to generate these images.
- The rolling log files for Ruby have been limited to current with 19 backups.
- The download manager may now use the file: protocol. This is useful to ingesting file into your archive from another archive.
- **apsRSDB.rb** program has a new option `--vical` for extracting vicarious calibration data from data base into a CSV file.
- **apsRSDB.rb** program has a new option `--purge` which can be used to manually purge data more than `n` days old.
- **apsRSDB.rb** program has a new option `--unlock` which can be used to remove a lock on a file.
- An error in the SQL command for inserting flag information into `rsdb_flags` table was fixed.
- An error for inserting number of pixels into `rsdb_flags` table was fixed.
- Updates to database tables.
  - the **aeronet_data** table now contains fields for date, time, and instrument. The table column order was also modified.
  - the views for MODIS L1B data now contains a column for MODIS L1A file from which it was derived.
  - links between calval tables and satellite tables were improved.
  - new validation tables (**validate_goci**, **validate_modisa**, and **validate_viirsn**).
- Ruby has been updated to v2.2.5-p239
- new NAVOCEANO SST operational coefficients.
- Updates to the vicarious calibration script (vc-aeronet.rb).
  - site location information read from database if not present in XML configuration file.
  - add units to axis and estimate of distance between satellite data and insitu data.
  - new resolution option for processing 250m MODIS from L1A input.

- ability to use ASD data as insitu-data source.
- ability to auto insert semi-screened satellite "insitu" data as calval cross-calibration input.
- auto-lock and tag files which produce good vicarious calibration results.
- differentiate between calibration (g00) and validation data (others).
- added aer_num_iter product to vicarious calibration outputs.
- added percent difference for validation to vicarious calibration outputs.
- insert validation data.
- improve logging esp. in a threaded environment.

- Several core libraries and tools have been updated.
  - gsl v1.16 replaces gsl v1.15
  - FOP has been updated to v2.0 from v1.1
  - doxbook-xsl-ns has been updated to v1.79.0 from v1.78.1
- A new vicarious calibration for VIIRS data is based on MOBY, Venise, and WaveCIS_Site_CSI_6 data June 2014 to August 2015. This calibration was performed in fall of 2015. It is designed for post C=0 VIIRS SDR data.

Table A.2. VIIRSN Vicarious Calibration

	410	**443**	**486**	**551**	**671**	**745**	**862**
v6.4	0.9798	0.9864	0.9813	0.9720	0.9686	0.9800	1.0000

Release v6.2.1

This version was released on 22 October 2015.

- New directory structure to benefit data layout.
- New start/stop options for map production (Level-3).

Release v6.2.0

This version was released on 10 September 2015.

- Updated n2gen to Goddard l2gen v8.1.4 (from v7.0.1). This corresponds to SeaDAS 7.2
  - correct OCI algorithm (ramp range updated)
  - use temperature/salinity correct aw, bbw in NIR iteration
  - new cirrus_opt cloud correction
  - remove all depenence on NBANDS
  - fix n2gen to allow NASA netCDF output using ofmt=25.

    - complete re-written MODIS IO code (from Goddard)
    - increased solz threshold for par to 90
    - added support for Raman scattering correction to GIOP
    - modifications to handle library reorganization (new libdfutils, libhdf4utils, libhdf5utils, libnetcdfutils, libtimeutils, removing libhdfutils)
    - updated calcite code to remove the 4/1.628 factor
    - Corrected many issues with the initial implementation of the Ahmad MSEPS aerosol selection scheme.
    - implement awhite interpolation within whitecaps.c
    - added support for OLI GEO file
- the MODIS cross-calibration file was updated (to 38d).
- the MODIS L1A to L1B LUTS were updated to V6.1.35.1_OC2.
- new AERONET-OC region (Galata_Platform)
- remove AERONET-OC region (Abu_Al_Bukhoosh). Also remove BATS and OregonInsitu.
- Updates to database tables.
    - the **rsdb_nrll2** table now contains image size information (samples,lines) and orbit number.
    - the **vicals** table now contains a hash of the csv column (minus ProcessingDate). This is used to handle reprocessing that might indicate a change in results. Additionally the columnds **sensor_id** and **epoch** were added for improved searching.
    - add OLCI sensor to **rsdb_sensors** table.
    - several new views have been added including: **view_level2**, **view_modisa_l1a**, **view_modisa_geo**, **view_modisa_1km**, **view_seawifs_l1a** and **view_viirs_sdr**.
    - new table **calval_daily_counts** which is used to quickly create a calendar indicating the number of aeronet readings per day.
    - new tables and views for satellite pixels (**rsdb_pixels**, **rsdb_pixels5**, **view_sat_pixels**, **view_sat_flags**,**view_sat_box3**, **view_sat_box3_mean**, **view_sat_box3_stdev**, **view_sat_box3_cv** **view_sat_box5**, **view_sat_box5_mean**, **view_sat_box5_stdev**, **view_sat_box5_cv**).
- Additional processing configuration options.
    - the option **realtime** can be used to make the database queries add time limits. If the time is a positive number, then it represents the number of seconds backwards to process. For example, `<realtime>172800</realtime>` will only process data for the past two days.

      If the value is negative, then it represents the UNIX epoch time of the time for which all prior data will be processed. For example, to process all data prior to Jan 1st 2015, the value 1420045200 will be used. That is, `<realtime>-1420045200</realtime>`.

      The GNU date command can be used to compute the UNIX epoch file. For example: `date --utc --date='2015-01-01T00:00:00' +%s`

- the option **email** can be used to force all error messages to this address rather than the default (whoami@hostname).

- Updates to vicarious calibration script.
  - the Encapsulated Postscript **eps** files have been replaced with Scalar Vector Graphs **svg** which are more web friendly.
  - add 5x5 box data to **rsdb_pixels** and **rsdb_pixels5**
  - process images to L2 even when no in-situ data available for producing the vicarious calibration runs.
  - improved processing from MODIS L1A.
- fix issue with imgMap for new VIIRS SST (VSSTO) file handling
- Ruby has been updated to v2.2.4-p175
- improve the automatic processing of L2 files to L3 files.
- improve the automatic processing of L4 composites.
- add --fs-move option to apsRSDB.rb to move files through the database from one directory to another. It performs a copy and then prints a series of rm commands for user to issue when ready.
- Updates to **imgMean**.
  - the `-k` and `-l` options now have reversed meaning. This was due to the relaxation of the requirement of having `l2_flags` on the command line.
  - the maximum number of files is now obtained from the operating system. This value can be found using the command `ulimit -n`.
  - the new auto-name creation was updated for the new APS file naming scheme. The output file is automatically named when the `-o` option is not used. For non-APS inputs, it is still best practice to use the `-o` option.
  - fix the flag count option which was not working.
- A new vicarious calibration for VIIRS data is based on MOBY data June 2014 to August 2015. This calibration was performed in September of 2015. It is designed for post C=0 VIIRS SDR data.

Table A.3. VIIRSN Vicarious Calibration

	410	443	486	551	671	745	862
v6.2	0.9807	0.9887	0.9823	0.9683	0.9655	0.9800	1.0000

- The vicarious calibration for MODIS data is based on MOBY data January 2013 to July 2015. This calibration was performed in July of 2015.

Table A.4. MODIS Vicarious Calibration

	412	443	469	488	531	547	555	645	667	678	748	859	869	1240	1640	2130
v6.2	0.9769	0.9879	1.0168	0.9923	0.9979	0.9984	1.0033	1.0270	0.9993	0.9989	0.9997	1.0184	1.0000	1.0000	1.0000	1.0000

- The following table shows the processing version number of each sensor as processing by APS v6.2.

Table A.5. Processing Version (APS v6.2)

Sensor	APS v6.2
goci	v04
modis	v14
viirs	v05

Release v6.0.0

This version was released on 21 April 2015.

- update imgMap to produce an averaged product ("bin").
- new global processing for MODIS and VIIRS creating 11km averaged ("bin") products.
- GOCI time correction due to slot acquisition time difference.
- update configuration to make single output files in *HDF v5* format.
- drop `true_color` product for `rhos` products. The true_color images are still created.
- use `cloud_eps` to try to recover cloud flagged pixels from simple threshold test (`rhos_865 > albedo`). For MODIS and GOCI processing only. VIIRS uses the VIIRS Cloud Mask product.
- The vicarious calibration for VIIRS data is based on MOBY data June 2012 to November 2014. This calibration was performed in December of 2014. It has been in place since APS v5.8.

Table A.6. VIIRSN Vicarious Calibration

	410	443	486	551	671	745	862
v6.0	0.9711	0.9769	0.9729	0.9564	0.9587	0.9800	1.0000

- The vicarious calibration for MODIS data is based on MOBY data June 2012 to June 2014. This calibration was performed in August of 2014. It has been in place since APS v5.6.

Table A.7. MODIS Vicarious Calibration

	412	443	469	488	531	547	555	645	667	678	748	859	869	1240	1640	2130
v6.0	0.9748	0.9875	1.0173	0.9921	0.9995	1.0003	1.0003	1.0259	0.9988	0.9975	0.9989	1.0254	1.0000	1.0000	1.0000	1.0000

- The vicarious calibration for GOCI data is based on a cross-calibration with MODIS data from the GOCI_CalSite_1 from 2011 to 2014. This calibration was performed in September of 2014. It has been in place since APS v5.6.

Table A.8. GOCI Vicarious Calibration

	412	443	490	555	660	680	745	865
v6.0	0.9676	0.9530	0.9173	0.8786	0.8807	0.8580	0.9430	1.0000

- The following table shows the processing version number of each sensor as processing by APS v6.0.

Table A.9. Processing Version (APS v6.0)

Sensor	APS v6.0
goci	v04

Sensor	APS v6.0
modis	v13
viirs	v04

- append many CF-compliant attributes to output file.
- set read-only permissions for all Level-1 data to avoid accidental removal.
- update **rsdb_fstats** field **perms** to character data
- update **rsdb_flags** adding new column **TRIMPIXEL** to hold TRIMPIXEL flag counts.
- new **rsdb_global** table
- fix **detnum** and **mside** extraction in vi-cal processing.
- new browse image XML structure (`etc/l2_browse.xml`, `etc/l3_browse.xml`, `etc/l4_browse.xml`).
- replace vc-aeronet PLIST configuration to new XML configuration structure (`etc/vc-aeronet.xml` replaces `etc/vc-aeronet.plist`).
- vical no longer produces the `.plist` file in `browse/jpss/aeronet/...`.
- vical CSV data version has been changed to “6.0”.
- VOCCO processing has been removed.
- Ruby has been updated to v2.2.2-p88
- the default list of L4 flags is now **CLDICE,LAND,ATMFAIL,MAXAERITER,HISOLZEN,HISATZEN,NAVWARN,NAVFAIL**.

Release v5.8.4

This version was released on 25 February 2015.

- fix issue with history product creation with HDF5 files
- add new options to **apsArea** so that it may complete faster depending upon the input file.
- Ruby has been updated to v2.2.0-p83
- refactor `regions.xml` processing to add bounding box to speed up coverage computation

Release v5.8.3

This version was released on 10 February 2015.

- update GOCI processing default albedo/cloud_eps
- update GOCI for true_color/NDVI in composites
- Ruby has been updated to v2.2.0-p43

Release v5.8.2

This version was released on 30 January 2015.

- update **rsdb_fstats** field size to DECIMAL(11,0) to handle 4GB+ files
- update GOCI processing to APS/hdf5 format
- Ruby has been updated to v2.2.0-p34
- improve database access and connectivity

Release v5.8.1

This version was released on 5 January 2015.

- data/common/product.xml - updates to several products
- new <dbscan><levels></levels></dbscan> configuration option
- new support of Scientific Linux v7
- Ruby has been updated to v2.1.5-p278

Release v5.8.0

This version was released on 29 December 2014.

- Updates to Ruby
  - Ruby has been updated to v2.1.5-p273
  - The ruby-prawn package is now supported as gem
- Updates to the data base
  - the **calval_site** table now uses DECIMAL(6,3) for lat/long rather than REAL.
  - the **calval_data**, **calval_dataw**, **calval_name** and **calval_wavelength**tables have been removed. Their fields have been inserted into the **calval_main** table.
  - the data base is supported by MySQL, Postgres, and SQLite.
- Updated n2gen to Goddard l2gen v7.0.1 (from v6.7.1)
  - updated bbw terms based on Zhang et al. (2009)
  - updated GIOP algorithm (using temperature/salinity correct bbw and nw).
  - new IOP algorithm (SWIM) **iop_opt=8** (the LMI algorithm is now set with**iop_opt=9**).
  - updated PAR algorithm.
  - set input file name to realpath of **ifile=8**. This allows input to be a symbolic link.
  - allow glob of VIIRS data to handle multiple granules on disk.
- Updated data files
  - updated land mask file **watermask15ARC.nc** (not used by MODIS which retains usage of **MOD44w.h5**).

  - updated met climatology file **met_climatology_v2014.hdf**.
  - updated ozone climatology file **ozone_climatology_v2014.hdf**.
  - new ancillary correction file **anc_cor_file_28jan2014.nc** for toms-like ozone with new **anc_cor_file=** keyword in **n2gen**.
  - updated NO2 climatology file **no2_climatology_v2013.hdf**.
  - new elevation file **ozone_climatology_v2014.hdf** with new **elev=** keyword in **n2gen**.
- the **aps.rb** has a new option **install** used to install any required Ruby gems. Currently, this includes only **prawn**.
- the default list of L4 flags is now **CLDICE,LAND,ATMFAIL,ATMWARN,MAXAERITER,HISOLZEN,HISATZEN,NAVWARN,NAVFAIL** .
- land processing is now available so that L4 Daily composites may have true color overlaid on image. The **-Y file=LAND** can be used to set this.
- many improvements when purging files and images from database. Better handling of deletion of entries in various tables to maintain integrity of data base.
- improvements to n2gen for handling input files **ifile=PATH** where the path is a symbolic link as well as the file globbing for VIIRS data that have multiple granules files.
- New vicarious calibration for VIIRS data based on MOBY data June 2012 to November 2014. This calibration was performed in December of 2014.

Table A.10. VIIRSN Vicarious Calibration

	410	443	486	551	671	745	862
v5.8	0.9711	0.9769	0.9729	0.9564	0.9587	0.9800	1.0000
v5.6	0.9739	0.9788	0.9743	0.9560	0.9606	0.9800	1.0000
v5.4	0.9797	0.9854	0.9852	0.9797	0.9862	1.0000	1.0000

- Processing Version Updates.

Table A.11. Processing Version (APS v5.8)

Sensor	APS v5.6	APS v5.8
goci	v03	v04
meris	v07	v07
modis	v12	v13
seawifs	v10	v10
viirs	v03	v04

Release v5.6.0

This version was released on 24 September 2014.

- New vicarious calibration for VIIRS data based on MOBY data June 2012 to June 2014. This calibration was performed in September of 2014.

Table A.12. VIIRSN Vicarious Calibration

	410	**443**	**486**	**551**	**671**	**745**	**862**
v5.6	0.9739	0.9788	0.9743	0.9560	0.9606	0.9800	1.0000
v5.4	0.9797	0.9854	0.9852	0.9797	0.9862	1.0000	1.0000

- New vicarious calibration for MODIS data based on MOBY data June 2012 to June 2014. This calibration was performed in August of 2014.

Table A.13. MODIS Vicarious Calibration

	412	**443**	**469**	**488**	**531**	**547**	**555**	**645**	**667**	**678**	**748**	**859**	**869**	**1240**	**1640**	**2130**
v5.6	0.9748	0.9875	1.0173	0.9921	0.9995	1.0003	1.0003	1.0259	0.9988	0.9975	0.9989	1.0254	1.0000	1.0000	1.0000	1.0000
v5.4	0.9731	0.9910	1.0132	0.9935	1.0002	0.9994	1.0012	1.0280	0.9996	0.9998	0.9989	1.0254	1.0000	1.0000	1.0000	1.0000

- New vicarious calibration for GOCI data based on cross-calibration with MODIS data from the GOCI_CalSite_1 from 2011 to 2014. This calibration was performed in September of 2014.

Table A.14. GOCI Vicarious Calibration

	412	**443**	**490**	**555**	**660**	**680**	**745**	**865**
v5.6	0.9676	0.9530	0.9173	0.8786	0.8807	0.8580	0.9430	1.0000
v5.4	0.9862	0.9753	0.9473	0.9149	0.9245	0.9223	0.9430	1.0000

- Update n2gen to Goddard l2gen v6.7.1 (from v6.6.7).
  - set NAVWARN when MODIS > 1 degree off nominal pointing.
  - deleted tsm_clark, poc_clark, chl_clark products.
  - new common sensor table for PIC.
  - new MLR coastal cdom products.
  - new Landsat8 OLI processing.
  - new HICO processing (using Goddard's processing code).
  - new XML-based method to define products (`data/common/product.xml`).
  - removal of SWIRNIR and NAAPS atmospheric correction codes.
  - fix regions_file parameter processing.
- Add new <timeframes> node in Level-3 generation configuration file to limit map creation to certain time frames.
- Add `regions_file` parameter for any Level-2 file that covers an AERONET-OC location. This option is enabled by adding a <params> XML node and the “areas” attribute of its sibling <param> to the `$APS_ETC/l2_gen.xml`.
- Allow processing of NASA/OBPG VIIRS data that has no f-correction applied. To use this data, the proper NASA/OBPG calibration file must be used via calfile= parameter.
- Updates to the visibility products based on AOPS processing codes.

  - the **imgBrowse** visibility color table was updated to NAVOCEANO version.
  - new **lmi_aph_g**, **lmi_adg_s**, **lmi_bbt** parameters for setting LMI coefficients.
  - new **hvis_iop** and **vvis_iop** parameters to determine possible input.
  - wavelength usage changes in LMI IOP algorithm.
  - selectable (sensor) wavelength for visibility.
  - ability to select non-sensor wavelength 532 visibility when using LMI IOP algorithm
  - LMI IOP algorithm updates from Paul Lyon
- Correct VIIRS mside product which was not being set. The mirror normal has been corrected (it was reversed).
- New region of interest for GOCI calibration (GOCI_CalSite_1).
- Update database tables.
  - the **rsdb_images** table now contains time information about when entry was created, last updates, and how many times.
  - new **vicals** table has been added.
  - new **vical_headers** table has been added.
  - new **rsdb_timeseries** table has been added.
  - the **rsdb_main** table now has an index on timeframe_id and ftype.
  - the **rsdb_aois** table now have uniqe constraints on name and md5sum.
  - introduce a mutex lock in RSDB.rb when calling high-level inserts of data files.
- Updates to the vicarious calibration methods:
  - the **satTime** is computed from center scan line time instead of starting line.
  - the **nLw_mean** and **chlor_a_mean** products are added to the CSV outputs.
  - the **AOC_Windspeed** is added to the CSV outputs for AERONET-OC matchups.
  - the **AOC_AOT** is added to the CSV outputs for AERONET-OC matchups.
  - the **vcal_gain** and **vcal_offset** products are added to the CSV outputs.
  - the **modmin** and **modmax** products are replaced with NASA standard names **aer_model_min** and **aer_model_max**.
  - there are several VIIRS specific unity gains with varying 750 channel calibrations (0.96,0.97,..1.04).
  - the AERONET-OC level 2.0 data is preferred over the 1.5 data (when available).
  - the calval database has been updated to include AERONET-OC 1.5 and 2.0 level data as of 16 September 2014.

- the MOBY filter-spec database has been updated for deployments 249 to 254 for post-deployment calibration (June 2014).
- the calval database properly adds the F0 term for MOBY data processing (nLw).
- support for MODIS calibration points
- the **calval_date** table now has an index on day, month, and year.
- the **calval_main** table now has several indicies.
- the **calval_instrument** table is not unique on instr_name and intr_type. Previously, it was unique only on instr_name.
- the **calval_level** table has been removed.
- the **site_code** of the **calval_site** table has been removed.

- New Level-4 (composite) processing. The following are available:
  - Daily Composite
  - Weekly Composite
  - Monthly Composite
  - 8-day running Composite
- All AERONET-OC related regions have been renamed to match the AERONET-OC offical names. For some, this means 'AAOT' becomes 'Venise'. For others, there is not change. LISCO and MVCO are both unchanged.

Table A.15. GOCI Vicarious Calibration

Old Name	New Name
AAOT	Venise
AbuAlBukhoosh	Abu_Al_Bukhoosh
Cove	COVE_SEAPRISM
Eureka	USC_SEAPRISM
Gageocho	Gageocho_Station
GDAT	Gustav_Dalen_Tower
Gloria	Gloria
GOT_Seaprism	GOT_Seaprism
Ieodo	Ieodo_Station
Helsinki	Helsinki_Lighthouse
LISCO	LISCO
LucindaJetty	Lucinda
MVCO	MVCO
Palgrunden	Palgrunden
WaveCIS	WaveCIS_Site_CSI_6

- Several core libraries have been updated.
  - HDF5 v1.8.12 replaces HDF5 v1.8.11
  - HDF v4.2.10 replaces HDF v4.2.9
  - netCDF v4.3.1.1 replaces netCDF v4.3.0
  - GeoTiFF v1.4.0 replaces GeoTiFF v1.2.2
  - PNG v1.6.9 replaces PNG v1.4.11
  - roxml v2.3.0 is a new dependency
  - lapack v3.4.2 is a new dependency
  - levmar v2.6 is a new dependency
  - Ruby has been updated to v2.1.2-p241
- Processing Version Updates.

Table A.16. Processing Version (APS v5.6)

Sensor	APS v5.4	APS v5.6
goci	v02	v03
meris	v07	v07
modis	v11	v12
seawifs	v10	v10
viirs	v02	v03

Release v5.4.0

This version was released on 19 February 2014.

- Several updates to Ruby:
  - Ruby was updated to v2.1.0-p30
  - ruby-prawn was updated to latest release (v0.14)
  - ruby-log4r was updated to latest release (v1.1.11)
  - ruby-pdf-reader was updated to latest release (v1.3.3)
  - Documentation updates
  - RSFile class was renamed RSFileScanner to avoid conflict with C extension class with same name
  - All Classes and constants in lib/aps, ext/aps, and ext/apshdf moved to APS module
  - APSHDF module removed
  - APS::QAA class updated to use QAA v6
  - Remove many compile time warnings

- The horizontal and vertical diver visibility products now use the LMI inherent-optical properties algorithm and provide the results at only the 532 nm wavelength regardless of input sensor.
- Update MODIS cross-calibration file to 26f.
- Update MODIS LUTs to v6.1.19_OC.hdf.
- Change the file transfer database tables. The **process** and **aps_dir** columns have been deleted.
- Fix day of year for VOCCO/VSSTO EDR products.
- Add option to imgMap to change way it handles control point grids. Use this new option of MODIS processing. Add new control points to MODIS L2 files that produce grids on the scan boundaries.
- Changes to GOCI processing:
  - Install new GOCI vicarious calibration values obtain from Wang. These are proxy values that improve estimates but are not final.
  - Fix GOCI processing for a subsection of the original image.
  - Fix GOCI map zoom factor to produce full 500m resolution maps.
  - Set GOCI version number to 2.
- Produce a water mass classification file (_TSS) for all sensors.
- Have imgMap count up flags for l2_flags array and store this information in the database.
- Fix processing of the OSI-SAF algorithm.
- Update n2gen so that a VIIRS OCC EDR file might be used as input. Since this data is already atmospherically corrected, the follow on products like chlorophyll-a, salinity, diver visibility, QAA may be generated.
- Expand the APS Level-2 input file to n2gen to handle GOCI, MERIS, MODISA and SeaWiFS files. The APS Level-2 file must contain the nLw products (as float) with the solar/sensor angles (azimuth and zenith). The latitude and longitude arrays must also be present.
- Several updates to vicarious calibration:
  - set channel vicarious calibration for all channels
  - use the Lwn f/Q product for AERONET-OC vicarious calibration
  - handle issue where image is not full box size
  - bound 865 channel for AERONET to 0.05. Negative values are set to 0.001.
  - various minor bug fixes
- Processing Version Updates.

Table A.17. Processing Version (APS v5.3)

Sensor	APS v5.0	APS v5.3
goci	v01	v02

Sensor	APS v5.0	APS v5.3
meris	v07	v07
modis	v11	v11
seawifs	v10	v10
viirs	v02	v02

Release v5.2.6

This version was released on 27 December 2013.

- E-book documentation now EPUB v3.
- Ruby documentation now available.

Release v5.2.5

This version was released on 17 December 2013.

- Minor bug fixes.
- New regions for processing: Ieodo and GulfofHonduras.

Release v5.2.4

This version was released on 6 December 2013.

- Minor bug fixes.

Release v5.2.3

This version was never officially released.

Release v5.2.2

This version was released on 24 November 2013.

- The APS processing was re-configured to run on separate machines with different data base search schemes. Therefore, host einsteinium is able to process the GOCI data alone. The host fermium is processing VIIRS data and host californium is processing MODIS/AQUA data. The host radon is handling VOCCO, data transfer, data import, vicarious calibration, and SAVANT processing.

  All newly written data is going to /rsng/lvl2 and /rsng/lvl3, and /rsng/browse. In several cases, the data are written to host-sensor specific RAIDS. For example, einsteinium has a 10TB RAID to hold all processed GOCI data. The MODIS data is held on a RAID on californium; VIIRS data is held on a RAID on fermium. The web browser has a new RAID to hold all images.

  The VIIRS data source has been moved to CLASS as oppossed to NAVO. This will provide larger VIIRS granules since CLASS aggregates the data by 4 granules. New CLASS subscriptions have been setup and data transfer is operational.

- New --tree option to apsRSDB.rb that shows Level-1 file history.

Release v5.2.1

This version was released on 5 November 2013. It fixed a major bug in v5.2.0 that prevented processing.

Release v5.2.0

This version was released on 1 November 2013.

- The APS processing has become entirely database dependent. The L1 to L2; L2 to L3; L3 to L4; creation of browse images from L2, L3, and L4 files; the maintanence of the vicarious calibration; and the testing of L1 file coverage is all done automatically though the use of SQL queries to the database.

  Therefore, the $APS_IN directory has been removed and will no-longer be part of APS. Instead, data will be acquired and automatically processed through: automated download (etc/downloads.xml); importation (via the import directory); or registation of a file with APS (apsRSDB.rb -a <file>).

- The vical_lines of SAVANT will now contain empty records whenever a file has been tested for each aeronet site (and MOBY). The empty record provides us with the answer of whether this file has been processed. These records correspond to files that cover aeronet region for which there is no available aeronet data.

- The RSDB table's **rsdb_mod1km**, **rsdb_modhkm**, and **rsdb_modqkm** have been updated to include a new field **lvl1_id** that corresponds to the L1A file that was used as input to the L1B file. This knowledge is used to determine which L1A files need to be processed to L1B.

  The RSDB table **rsdb_fstats** has become a child table of **rsdb_main** using the **main_id** as key. The original table was indpendent having its own id. Thus, the field **fstat_id** has also been removed from **rsdb_main**.

- The VIIRS Cloud Mask (VCM) default mask value was reset to 1. This means any pixel deemed “Probably Clear”, “Probably Cloudy”, or “Confidently Cloudy” will have **CLDICE** flag set. This makes the cloud mask more aggressive.

- The program apsArea was updated to use navigation from GOCI L1B file directly.

- The CalValDB was updated to work with SQLite and obtain AERONET data given a URL. The computation of the **epoch** field in **calval_main** was corrected. Like, the TransferDB tables, the tables of the CalValDB are now part of the remote-sensing data base (RSDB).

  A new table **calval_scan** contains the date when a particular AERONET-OC site was last updated. This is used to trigger automatic refreshes. To further enhance this aquistion information, the **calval_main** table has a new field **added** to indicate when each record was added. Combined with the **calval_scan**, a query may be performed that returns all records since the last additions.

  The CalValDB database is used for the vicarious calibration instead of SAVANT.

- The vc-aeronet.rb program was modularized into a library module (**ViCal**). Most of the code was pushed to the ruby repository and is now located in `lib/aps/vical.rb`.

- The OSI-SAF SST coefficents were updated to the latest October 2013. These coefficents were obtained from NOAA. Additionally, the CMC 0.2 degree L4 SST product and the NAVO K10 product may be used as input (sstfile=).

- Add new AERONET-OC site Canberra.

- The netCDF library was updated to v4.3.0.

Release v5.1.0

This version was released on 1 October 2013.

- The SST processing was updated to add the OSI-SAF, NL45DE+2, and NL45DE algorithms. The product names for these are: sst_osisaf, sst_nl2_navo, and sst_nl_navo, respectively. The sst_nl_navo is updated from the previous version of APS. The sst_mc_navo is still available, however, the coefficients are out of date. The default sst algorithm is OSI-SAF.

  The MODIS sst algorithm has been renamed sst_miami. The default sst algorithm is the sst_miami algorithm.

  The VIIRS and MODIS processing generates a separate SST file which includes the brightness temperatures, sensor azimuth angles, the standard (default) SST algorithm, and the new SST algorithms (VIIRS only).

- The GOCI library was updated from SeaDAS 7.0.1 so that n2gen no longer needs the lat/lon file in data/goci directory (geofile does not need to be set).

- The vc-aeronet.rb added a "g02-std" for VIIRS only that processes using the WaveCIS vicarious calibration gains.

- The MODIS cross-calibration file was updated (to 26e).

- Several core libraries have been updated.

  - HDF5 v1.8.11 replaces HDF5 v1.8.10

  - HDF v4.2.9 replaces HDF v4.2.8

  - PROJ.4 v4.8.0 is a new dependency

Release v5.0.3

This version was released on 26 September 2013.

Minor bug fixes.

Release v5.0.2

This version was released on 5 August 2013.

- Updates to vicarious calibration:

  - There were some bugs when running vc-aeronet standalone outside of aps.rb.

  - The rrs_model program was updated to make the input spectrum as coming from AERONET and not MODIS as it was original programmed. The F0, aw, and bbw are currently proxy with 412, 443, 488, 555, 660, 860 channels using SeaWiFS values. The AERONET "sensor" is reduced to the 6 (of AERONET's original 8 -- a 530 and 1020 we do not need) channels that the hyperspectral Rrs code requires. A bug was fixed so that convolution with the RSR table was correct.

  - Since rrs_model will compute an nLw for each desired sensor from the aeronet band, the shift option was removed (we will _always_ shift). In a related note, the RSR table information was moved from its current location.

  - Since the uncertainity index is adding complexity with no return in results, the pui/mui stuff has been removed.

  - Change recipe handling so that we always run g00-std and g01-std, where g00-std runs the standard atmospheric correction with vicarious nLw inputs with gains=1.0 and offsets=0.0. With g01, the only difference is that the currently valid (data/<sensor>n2gen_defaults.par) vicarious calibration is used. This may be useful for a "check" to see how the ratios of vLt/Lt change -- it should not be used to generate a vicarious gain. That is the role of g00-std.

- Update the way data is queried from SAVANT using the 'epoch' field as opposed to the year, day of year. This should improve matchups around day break (MOBY) and year turn-over.
- Fix the header written out for the CSV files.
- Add a quality column to the give some indication of the quality of the particular matchup. This quality is based on l2_flags only.

Release v5.0.1

This version was released on 8 August 2013.

Minor bug fixes required to make v5.0.0 operational.

Release v5.0

This version was released on 6 August 2013.

- A new remote sensing data base. All data to be processed by this version of APS must be registered. All data produced by this APS will be registered. Many new capabilities are available with this new data base.
- Totally new configuration processing. This new processing allows the user to process data at each stage (lvl1, lvl2, lvl3) using new XML user configuration files. There are XML files for going from Level-1 to Level-2, Level-2 to Level-3, and Level-3 to Level-4. There are another set of XML configuration files that define how Level-2, Level-3, and Level-4 images are created.
- Much more fine-grained threading so that new programs are run in parallel. For example, several instances of **imgBrowse** may be running at the same time.
- New VIIRS OCC EDR processing from Level-2 to Level-3 and browse image generation.
- New Level-2 based data extraction to SAVANT procedure. This is for filling in the SAVANT satellite pixels data tables.
- A new default chlor_a algorithm: Ocean Color Index (OCI) by Hu.
- Updated GOCI Rayleigh/aerosol tables (from Goddard) and band-pass values.
- MERIS processing has been updated to V2012.0.
  - MERIS N1 inputs should be from the MERIS 3rd reprocessing.
  - new vicarious calibration values from NASA.
  - switch to 779 channel for atmospheric correction.
  - new coccolithophore calibration values.
- Updates to vicarious calibration processing:
  - GOCI data can now be processed.
  - The -g0- settings will always use a gain of 1.0 and intercept of 0.0.

  - The output vi-cal file is now larger for future additional tests. The vi-cal is still an extraction of the center matchup pixel

- Processing Version

Table A.18. Processing Version (APS v5.0)

Sensor	APS v4.8	APS v5.0
goci	v01	v01
meris	v06	v07
modis	v10	v11
seawifs	v09	v10
viirs	v01	v02

Appendix B. Acknowledgements

The following people have contributed to the development of APS while working for or at the Naval Research Laboratory at Stennis Space Center, MS.

- Paul Martinolich (Lead), Ocean Color processing (SeaWiFS, MODIS, MERIS, GOCI, and VIIRS), data base capabilities, vicarious-calibration capabilities, in-situ processing, image processing functions, imgBrowse, NAVOCEANO transitions, documentation
- Brandon Casey, imgBrowse - contours, curvy vectors, fonts, Google Earth, imgMean - latest pixel composites, imgSmooth, imgFillGaps
- Paul Lyon, MERIS
- Sherwin Ladner, Operations
- Adam Lawson, Operations and SAVANT capabilities